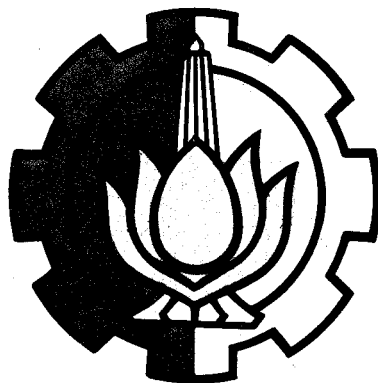


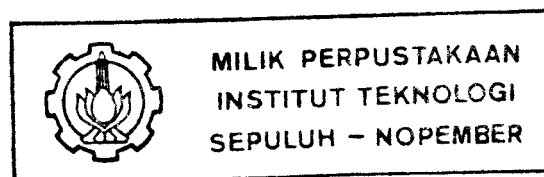
3100096007593

PERPUSTAKAAN ITS	
Tgl. Terima	22 SEP 1994
Terima Dari	H
No. Agenda Prp.	2735

ELEKTRONIK PENGATURAN DAN OTOMATISASI
PENCAMPURAN KOMPOSISI MINUMAN DENGAN
MENGUNAKAN MIKROKONTROLLER 8751H YANG
DAPAT DIHUBUNGKAN KE KOMPUTER IBM PC



RSE
621.391 6
Dur
2-1
1994



Oleh :

Darwison

NRP : 2902201624

JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
1994

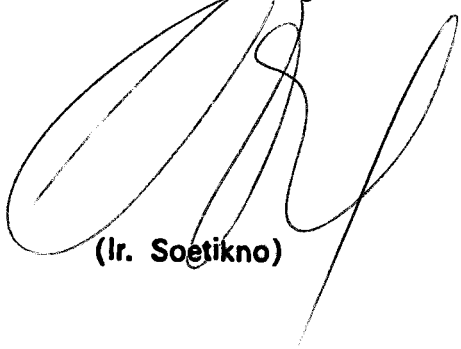
ELEKTRONIK PENGATURAN DAN OTOMATISASI
PENCAMPURAN KOMPOSISI MINUMAN DENGAN
MENGUNAKAN MIKROKONTROLLER 8751H YANG
DAPAT DIHUBUNGKAN KE KOMPUTER IBM PC

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar
Sarjana Teknik Elektro
Pada
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya

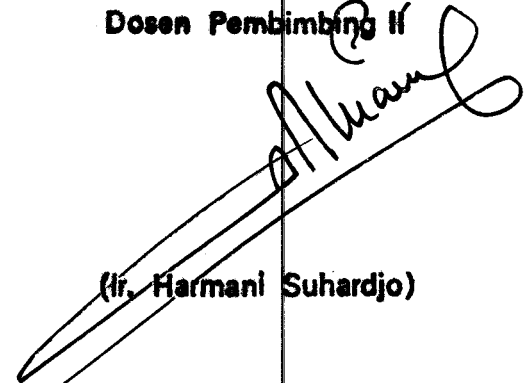
Mengetahui / Menyetujui :

Dosen Pembimbing I



(Ir. Soetikno)

Dosen Pembimbing II



(Ir. Harman Suhardjo)

SURABAYA
AGUSTUS, 1994

ABSTRAK

Menurunnya kualitas dan kuantitas suatu produk dapat disebabkan oleh faktor keterbatasan manusia yang diperkerjakan. Begitu juga dalam hal mencampur komposisi minuman yang dilakukan langsung oleh manusia. Faktor keterbatasan disini dapat dalam hal; membuat komposisi minuman baru, ketepatan mencampur dan kecepatan menghasilkan suatu komposisi minuman.

Berdasarkan masalah tersebut diatas maka dibuatkanlah suatu alat bantu yang terdiri dari mikrokontroller, komputer IBM PC, dan penunjang lainnya.

Pada tugas akhir ini akan dibuat suatu elektronik pengaturan dan otomatisasi pencampuran komposisi minuman dengan menggunakan mikrokontroller 8751H yang dapat dihubungkan ke komputer IBM PC.

KATA PENGANTAR

Atas berkat rahmat Tuhan Yang Maha Esa, maka penulis berhasil menyelesaikan perencanaan dan pembuatan alat serta penyusunan naskah Tugas Akhir yang berjudul :

**ELEKTRONIK PENGATURAN DAN OTOMATISASI PENCAMPURAN
KOMPOSISI MINUMAN DENGAN MENGGUNAKAN
MIKROKONTROLLER 8751H YANG DAPAT DIHUBUNGKAN KE
KOMPOTER IBM PC**

Tugas akhir ini adalah merupakan salah satu syarat yang harus ditempuh oleh setiap mahasiswa untuk meraih gelar kesarjanaan di Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Dalam mengerjakan Tugas Akhir ini penulis banyak mendapatkan bantuan dan bimbingan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

- Ir. Soetikno, selaku dosen pembimbing dan Koordinator Bidang Studi Elektronika, Jurusan Teknik Elektro, Fakultas Teknologi Industri ITS.
- Ir. Harmani Suhardjo, selaku dosen pembimbing.
- Ir. Murdi Asmoroadji, selaku dosen wali.
- Dr. Ir. Moch. Salehudin, M. Eng. Sc., selaku Ketua Jurusan Teknik Elektro, FTI ITS.
- Seluruh staf pengajar dan administrasi Jurusan Teknik Elektro, FTI ITS,

yang telah membantu kelancaran pelaksanaan Tugas Akhir ini.

- Seluruh rekan-rekan mahasiswa Jurusan Teknik Elektro Bidang Studi Elektronika dan semua pihak yang telah memberikan bantuan, baik secara langsung maupun tak langsung.

Akhir kata, penulis berharap semoga segala sesuatu yang telah dihasilkan dalam pelaksanaan Tugas Akhir ini dapat bermanfaat bagi kemajuan ilmu pengetahuan dan kesejahteraan umat manusia.

Surabaya, Agustus 1994

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
ABSTRAK	iii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
BAB I	
PENDAHULUAN	I-1
I.1. Latar Belakang	I-1
I.2. Permasalahan	I-2
I.3. Pembatasan Masalah	I-2
I.4. Tujuan	I-2
I.5. Metodologi	I-3
I.6. Sistematika	I-3
BAB II	
TEORI PENUNJANG	II-1
II.1. PENDAHULUAN	II-1
II.2. ISTILAH DALAM MINUMAN	II-1

II.2.1. Long Drink	II-1
II.2.2. Intilah Based	II-1
II.2.3. Liquer	II-1
II.2.4. Cocktail	II-2
II.2.5. Pendekatan Teknologi Komputer dan Automatisasi Dalam Mencampur Minuman.	II-4
II.3. URAIAN UMUM MIKROKONTROLLER CHIP TUNGGAL	II-5
II.3.1. ARSITEKTUR INTERNAL 8751H	II-6
II.3.2. FUNGSI TIAP PIN INTEL 8751H	II-6
II.3.3. HUBUNGAN KERJA ANTAR SUBSISTEM	II-10
II.3.4. CENTRAL PROCESSING UNIT (CPU)	II-11
II.3.5. MEMORI	II-15
II.3.5.1. Read Only Memory (ROM)	II-16
II.3.5.2. Random Access Memory (RAM)	II-17
II.3.6. PEWAKTU/PENCACAH	II-18
II.3.7. SISTEM INTERRUPT	II-20
II.3.8. PORT SERIAL	II-22
II.4. KOMUNIKASI DATA SERIAL	II-24
II.4.1. KONEKTOR DB-25	II-25
II.4.2. FORMAT DATA	II-26
II.4.3. INTERKONEKSI KABEL	II-27
II.5. MOTOR STEPPER	II-29

II.5.1. Variable Reluctance Motor	II-30
II.5.2. Permanent Magnet Motor	II-31
II.5.3. Permanent Magnet Hybrid Motor	II-34

BAB III

PERENCANAAN PERANGKAT KERAS	III-1
III.1. PENDAHULUAN	III-1
III.2. DIAGRAM BLOK	III-1
III.3. RANGKAIAN MIKROKONTROLLER 8751H	III-3
III.4. RANGKAIAN DEKODER MIKROKONTROLLER 8751H	III-4
III.5. RANGKAIAN MASUKAN/KELUARAN	III-5
III.6. RANGKAIAN PENGKONDISI SINYAL	III-5
III.7. RANGKAIAN SERIAL INTERFACE RS-232C	III-9
III.8. RANGKAIAN MOTOR STEPPER	III-11

BAB IV

PERENCANAAN PERANGKAT LUNAK	IV-1
IV.1. PENDAHULUAN	IV-1
IV.2. Software untuk system 8751h	IV-1
IV.2.1. Proses Pemindahan Data	IV-1
IV.2.2. Pemetaan Memory Program	IV-2
IV.2.3. Instruksi Pemindahan Data	IV-2

IV.2.4. Instruksi Pada Pengoperasian Akumulator	IV-4
IV.2.5. Instruksi untuk Percabangan	IV-5
IV.3. PERANGKAT LUNAK MIKROKONTROLLER 8751h. . .	IV-7
IV.4. PERANGKAT LUNAK PADA IBM - AT	IV-9

BAB V

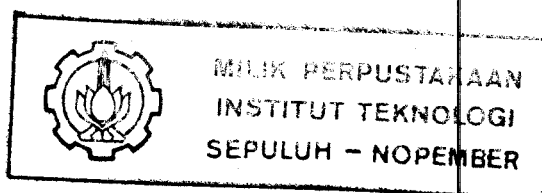
PENGUJIAN DAN CARA PENGOPERASIAN ALAT	V-1
V.1. PENDAHULUAN	V-1
V.2. PENGUJIAN KOMUNIKASI SERIAL	V-1
V.3. PENGUJIAN OUTPUT ALAT	V-2
V.4. PILIHAN FUNGSI PADA MENU UTAMA	V-3
V.5. CONTOH CARA MEMBUAT SUATU KOMPOSISI MINUMAN BARU	V-4
V.6. CONTOH CARA MENGOPERASIKAN ALAT	V-6

BAB VI

PENUTUP	VI-1
VI.1. KESIMPULAN	VI-1
VI.2. SARAN-SARAN	VI-2

DAFTAR PUSTAKA

LAMPIRAN



DAFTAR GAMBAR

GAMBAR	HALAMAN
2.1 Hot Today	II-2
2.2. Long Island Tea	II-3
2.3. Blok diagram proses dalam mencampur cocktail secara manual	II-4
2.4. Blok diagram proses dalam mencampur cocktail dengan menggunakan alat otomatis	II-5
2.5. Konfigurasi pin 8751H	II-8
2.6. Arsitektur Diagram Blok Keluarga MCS-51	II-12
2.7. Peta alokasi ROM	II-17
2.8. Peta Alokasi RAM	II-19
2.9. Transmisi Asinkron	II-25
2.10. Karakteristik Konektor DB-25	II-26
2.11. Hubungan Antara PC dengan PC	II-28
2.12. Prinsip dasar putaran motor stepper	II-30
2.13. Variable reluctance motor	II-32
2.14. Permanant magnet motor	II-33
2.15. Permanent magnet hybrid motor	II-34
3.1. Diagram blok dari alat yang direncanakan	III-2

3.2. Rangkaian Mikrokontroller 8751H	III-3
3.3. Rangkaian decoder	III-5
3.4. Rangkaian masukan/keluaran	III-6
3.5. Rangkaian pengubah sinyal taraf 12V ke TTL	III-7
3.6. Karakteristik Current transfer ratio 4N25	III-8
3.7. Rangkaian pengubah sinyal taraf TTL ke 12V	III-9
3.8. Perbandingan spesifikasi level RS-232C dengan TTL	III-10
3.9. Rangkaian pengubah sinyal	III-10
3.10. Hubungan Port Serial RS-232C	III-11
3.11. Rangkaian driver motor stepper	III-13
4.1. Flow chart program pada mikrokontroller 8751H	VI-8
4.2. Flow chart program komputer	VI-10
5.1. Tampilan Menu Utama di komputer IBM PC	V-5

DAFTAR TABEL

TABEL	HALAMAN
2.1. Komposisi Long Island Tea	II-3
2.2. Register TMOD	II-20
2.3. Alamat awal dari interrupt service program	II-20
2.4. Register SCON	II-23
3.1. Pemetaan memori dari I/O	III-4
3.2. Eksitasi motor stepper	III-12
4.1. Pemetaan memori data dalam	IV-7
5.1. Pengukuran volume dibandingkan dengan data volume dari IBM PC	V-2

BAB I

PENDAHULUAN

I.1. Latar Belakang

Sejalan dengan semakin meningkatnya kehidupan masyarakat metropolitan seperti di Jakarta, Surabaya dan kota-kota besar lainnya di dunia. Maka problema yang dihadapi juga semakin meningkat.

Sebagai akibatnya manusia memerlukan kompensasi dari jam-jam sibuk yang menguras kebugaran pikirannya. Beberapa dari mereka berusaha mencari secangkir minuman disamping sebagai pelepas dahaga juga supaya dapat menambah semangat kerjanya. Atau kalau diluar jam kerja mereka dapat mencari suasana santai dengan ditemani segelas minuman sambil mendengarkan alunan musik seperti di music lounge, bar, pub sehingga dapat melepaskan emosi jiwa yang dialaminya selama bekerja.

Melihat gejala ini maka harus diusahakan penyesuaian selera minuman pengunjung yang bervariasi dan terkadang sangat kritis. Dengan seorang waiter atau bartender sangatlah memainkan peranan penting untuk mewujudkan kreasinya dan menyesuaikan dengan selera konsumennya. Tetapi untuk mewujudkan itu semua tidaklah mudah, karena harus dicari orang-orang yang benar-benar profesional dalam bidangnya. Disamping bayarannya mahal juga masih sedikit orang-orang yang mempunyai profesi itu.

Untuk itulah dalam membantu kerja dari seorang waiter atau bartender dibuatkanlah suatu alat yang dapat meringankan kerja mereka.

I.2. Permasalahan

Menurunnya kualitas dan kuantitas suatu produk dapat disebabkan oleh faktor keterbatasan manusia yang dipekerjakan. Begitu juga dalam hal mencampur komposisi minuman yang dilakukan langsung oleh manusia. Aktivitas ini kebanyakan mengandalkan fisik dan berulang-ulang. Misalnya mengambil botol, meletakkan botol, menuangnya dan sebagainya. Selain itu juga adanya faktor keterbatasan manusia dalam hal: membuat komposisi minuman baru, ketepatan mencampur dan kecepatan menghasilkan suatu komposisi minuman. Dimana faktor ketepatan berkaitan dengan cita rasa dari suatu minuman. Sedangkan faktor kecepatan berkaitan dengan kualitas pelayanan dari pub, hotel dan lain sebagainya.

I.3. Pembatasan Masalah

Untuk mengatasi faktor keterbatasan manusia tersebut diatas maka dibuatkan suatu alat otomatis yang terdiri dari: washer pump untuk memompakan minuman ke pipa, empat valve dari empat macam minuman yang akan dicampurkan untuk membatasi volume minuman yang akan dikeluarkan, serta empat tabung minuman yang kesemuanya ini dikontrol oleh IC single chip mikrokontroller 8751. Juga dapat dihubungkan ke komputer IBM PC secara serial untuk dapat membuat komposisi minuman baru.

I.4. Tujuan

Tujuan dari Tugas Akhir ini adalah untuk mempelajari teknik

pengukuran volume minuman dari kecepatan aliran minuman dalam pipa. Dengan dibantu oleh IC single chip mikrokontroller 8751 maka dapat membantu dalam hal proses pengukuran volume minuman tersebut. Dan untuk pemakaian yang lebih baik dapat menggunakan komputer IBM PC.

I.5. Metodologi

Untuk merealisasi apa yang sudah direncanakan maka dibuatkan pengaturan secara software pada Mikrokontroller 8751H yang nantinya akan memberikan timer ke buka-tutupnya valve untuk volume minuman yang diinginkan.

Dimana volume minuman diset lewat komputer IBM PC sebagai komposisi minuman baru yang diinginkan.

I.6. Sistematika

Sistematika pembahasan dan penyusunan tugas akhir ini dilakukan sebagai berikut :

- BAB I : Pendahuluan , pada bab ini membahas tentang latar belakang, permasalahan, pembatasan masalah, tujuan, metodologi sistematika dan relevansi.
- BAB II : Teori penunjang, pada bab ini membahas tentang pengenalan minuman cocktail, teori umum mikrokontroller 8751H.
- BAB III : Perencanaan, pada bab ini membahas tentang perencanaan perangkat keras yang terdiri dari rangkaian mikrokontroller

8751H dan hubungan serial ke komputer IBM PC.

- BAB IV : Perencanaan perangkat lunak yang terdiri dari program pengoperasian minimum sistem dan program pengoperasian dengan dihubungkan ke komputer IBM PC. Yang kesemuanya menggunakan bahasa Assembly MCS-51 dan bahasa pascal.
- BAB V : Merupakan penutup yang berisi hasil pengamatan alat ini serta kemungkinan penyempurnaan dan pengembangan dari peralatan yang telah direncanakan.

BAB II

TEORI PENUNJANG

II.1 PENDAHULUAN

Teori yang menunjang dalam perencanaan perangkat keras maupun perangkat lunak dalam tugas akhir ini meliputi macam-macam minuman, mikrokontroller 8751H, hubungan serial ke IBM PC-AT.

II.2 ISTILAH DALAM MINUMAN

II.2.1 Long Drink

Suatu minuman dikatakan 'long drink' apabila tujuan dari minuman itu dibuat bukan sebagai pelepas rasa dahaga, melainkan untuk dinikmati sekaligus memberikan suasana yang diharapkan. Dibawah ini terdapat beberapa istilah yang erat hubungannya dengan pokok pembahasan ini.

II.2.2 Intilah Based

Istilah 'based' yang dimaksud disini adalah faktor rasa yang dominan dalam suatu formula. Yang tentunya disebabkan oleh dominasi dari segi jumlah volume. Tetapi tidak semua minuman dapat dijadikan base. Hanya minuman tertentu yang memiliki sifat base misalnya Bacardi-based, Vodka-based, Gin-based dan lain sebagainya.

II.2.3 Liquer

Liquer yang terkadang pula disebut spirit, menyajikan suatu bentuk minuman yang uniform. Biasanya minuman yang mempunyai sifat base dapat

dijadikan liquer.

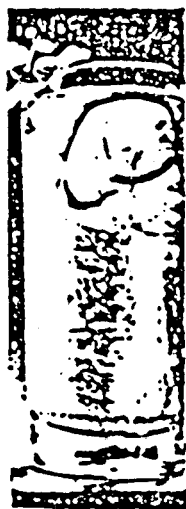
II.2.4 Cocktail

Dalam membuat cocktail dibutuhkan keahlian dalam mencampur segala jenis minuman yang dirasakan dapat membentuk cita rasa baru yang lebih spesifik.

Pencampuran cocktail disesuaikan dengan situasi yang dirasakan pada saat itu. Misalnya, Eye Openers, disaat permulaan dari siklus kehidupan di pagi hari, untuk membangkitkan spirit dan appetite; Nighthtcaps, mengantar pada proses relaksasi ; Party's Delight, memercikkan api gelora kegembiraan suasana pesta.



Gambar 2.1.
Hot Toddy, contoh sebuah cocktail yang tergolong nightcaps



Gambar 2.2.
Long Island Tea (vodka-based)

Sebuah contoh cocktail yang tergolong Afternoon Delight

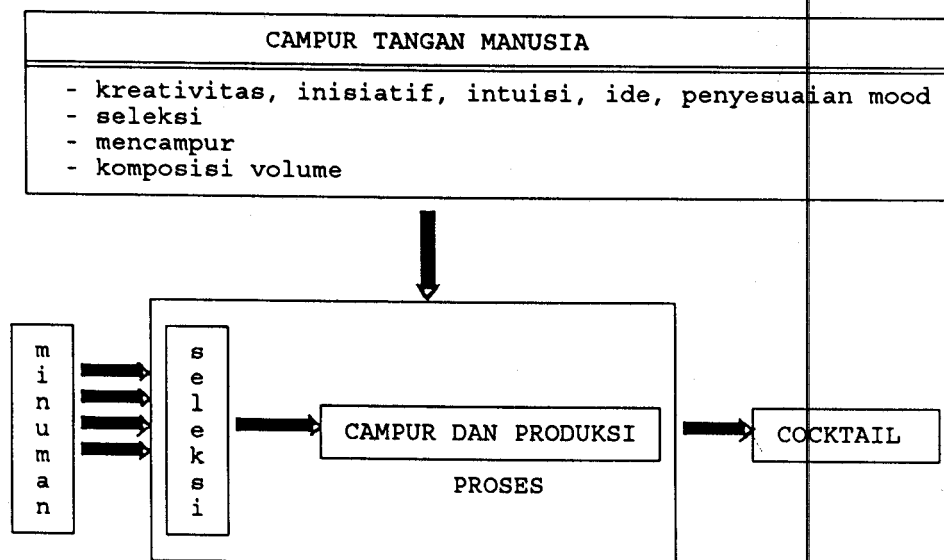
Tabel 2.1.
Komposisi Long Island Tea

INGREDIENTS	AMOUNT (ml)
Vodka	30
Teguila	30
Bacardi	30
Cointreau	15
Lemon Juice	15
Sugar Syrup	15
Cola	30

II.2.5 Pendekatan Teknologi Komputasi dan Automatisasi

Dalam Mencampur Minuman

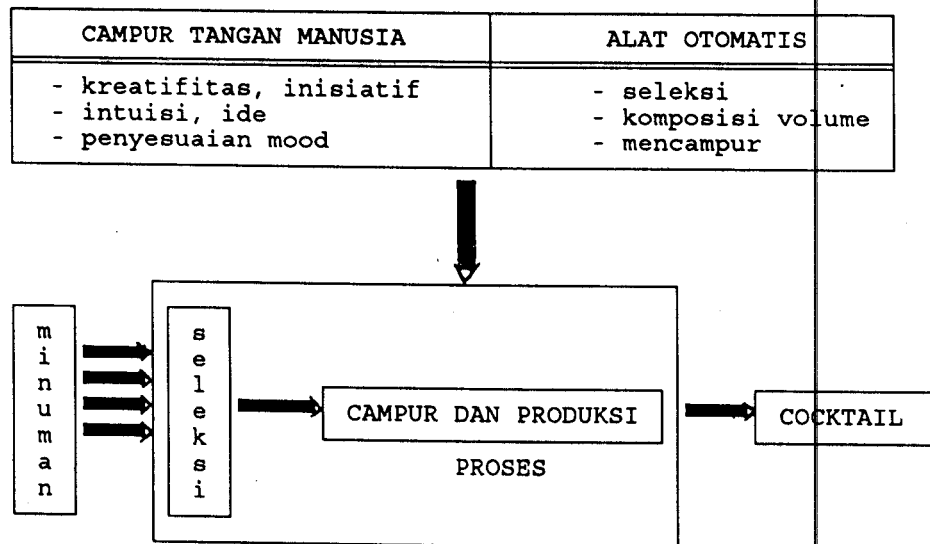
Pada Gambar 2.3. terlihat proses mencampur cocktail secara manual.



Gambar 2.3.
Blok diagram proses dalam mencampur cocktail secara manual.

Minuman-minuman adalah seakan-akan input berupa raw material dalam suatu proses. Dengan diberikan pengaruh luar dari manusia (human intervention) maka proses akan menghasilkan produk (dalam hal ini berupa cocktail) dengan sifat dan rasa yang diinginkan.

Gambar 2.4. terlihat proses dalam seni mencampur cocktail dengan menggunakan alat otomatis.



Gambar 2.4.

Blok diagram proses dalam mencampur cocktail dengan menggunakan alat otomatis

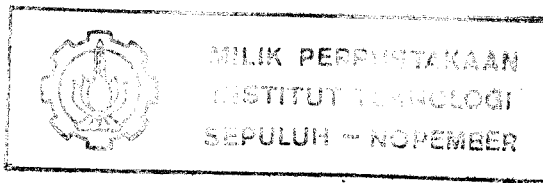
Dari gambar tersebut terlihat campur tangan manusia diminimumkan menjadi hanya tugas yang sifatnya lebih manusiawi yakni:

- Kreativitas
- Inisiatif
- Ide
- Intuisi
- Penyesuaian jenis cocktail yang dipilih dengan situasi yang sedang dirasakan saat itu (mood adjustment).

Sedangkan untuk tugas selanjutnya diserahkan pada alat otomatis dengan kualitas pengerjaan yang melebihi kemampuan seorang bartender.

II.3 URAIAN UMUM MIKROKONTROLLER CHIP TUNGGA

Dalam perencanaan peralatan ini digunakan komponen mikrokontroller



chip tunggal 8-bit dari keluarga Intel MCS-51 type 8751H.

Dasar pertimbangan pemilihan penggunaan komponen tersebut diatas adalah sebagai berikut :

1. Memiliki CPU 8-bit dan tersedia fasilitas instruksi yang diperlukan untuk pengaturan.
2. Memiliki EPROM internal sebesar 4096 byte (4 Kbyte).
3. Memiliki RAM internal sebesar 128 byte.
4. Memiliki 4 buah I/O Port, 32 jalur I/O.
5. Memiliki 2 buah internal timer/counter 16 bit.
6. Memiliki 5 buah sumber Interrupt.

Komponen mikrokontroller Intel 8751H mempunyai kompatibilitas yang tinggi dengan komponen peripheral yang dikeluarkan oleh INTEL atau dengan komponen TTL.

II.3.1 ARSITEKTUR INTERNAL 8751H

Komponen Intel 8751H menerapkan teknologi N-Channel Silicon Gate HMOS (High Speed Metal Oxide Silicon) dalam bentuk Integrated Circuit (IC) yang dikemas dalam bentuk DIP (Dual Inline Package) dengan jumlah pin 40. Konfigurasi pin pada Intel 8751H diperlihatkan pada gambar 2.5.

II.3.2 FUNGSI TIAP PIN INTEL 8751H

Konfigurasi pin Intel 8751H dikelompokkan sebagai I/O Port, pin-pin kontrol, pin Interrupt, pin Xtal dan pin Power Supply. Fungsi dari masing-masing pin adalah sebagai berikut :

- VSS

Merupakan pin ground.

- VCC

Dihubungkan dengan tegangan catu +5 Volt.

- Port 0

I/O Port 8-bit dengan open drain. Pada saat pemrograman dan perubahan (verifikasi) digunakan sebagai jalur input/output data. Jika menggunakan external memori, maka port ini berfungsi sebagai data bus dan address bawah (low address). Port 0 dapat menerima beban dua IC TTL.

- Port 1

Merupakan 8-bit quasi bidirectional port, juga digunakan sebagai alamat bawah (low address) pada saat dilakukan pemrograman dan perubahan.

- Port 2

Port 2 juga 8 bit quasi bidirectional port, juga digunakan untuk alamat atas (high address) pada saat menggunakan memory eksternal. Pada saat pemrograman pin ini digunakan sebagai alamat atas (high address) serta sinyal pengontrol. Port ini dapat menerima beban satu IC TTL.

- Port 3

I/O port quasi bidirectional yang juga mengandung pin Interrupt, timer, pin untuk serial port serta RD dan WR yang dapat digunakan untuk berbagai pilihan. Jika hasil output latch yang diinginkan adalah fungsi kedua, maka output latch harus diprogram pada kondisi 1 (high) sehingga fungsi tersebut dapat aktif. Port 3 dapat menerima beban satu TTL. Fungsi kedua dari port

Port 1 Bit 0	1	P1.0	Vcc	40	+5V	
Port 1 Bit 1	2	P1.1	(AD0)P0.0	39	Port 0 Bit 0	(Address/Data 0)
Port 1 Bit 2	3	P1.2	(AD1)P0.1	38	Port 0 Bit 1	(Address/Data 1)
Port 1 Bit 3	4	P1.3	(AD2)P0.2	37	Port 0 Bit 2	(Address/Data 2)
Port 1 Bit 4	5	P1.4	(AD3)P0.3	36	Port 0 Bit 3	(Address/Data 3)
Port 1 Bit 5	6	P1.5	(AD4)P0.4	35	Port 0 Bit 4	(Address/Data 4)
Port 1 Bit 6	7	P1.6	(AD5)P0.5	34	Port 0 Bit 5	(Address/Data 5)
Port 1 Bit 7	8	P1.7	(AD6)P0.6	33	Port 0 Bit 6	(Address/Data 6)
Reset Input	9	RST	(AD7)P0.7	32	Port 0 Bit 7	(Address/Data 7)
Port 3 Bit 0 (Receive Data)	10	P3.0(RXD)	(Vpp)/EA	31	External Enable	(EPROM Programming Voltage)
Port 3 Bit 1 (XMIT Data)	11	P3.1(TXD)	(PROG)ALE	30	Address Latch Enable	(EPROM Program Pulse)
Port 3 Bit 2 (Interrupt 0)	12	P3.2($\overline{\text{INT0}}$)	$\overline{\text{PSEN}}$	29	Program Store Enable	
Port 3 Bit 3 (Interrupt 1)	13	P3.3($\overline{\text{INT1}}$)	(A15)P2.7	28	Port 2 Bit 7	(Address 15)
Port 3 Bit 4 (Timer 0 Input)	14	P3.4(T0)	(A14)P2.6	27	Port 2 Bit 6	(Address 14)
Port 3 Bit 5 (Timer 1 Input)	15	P3.5(T1)	(A13)P2.5	26	Port 2 Bit 5	(Address 13)
Port 3 Bit 6 (Write Strobe)	16	P3.6($\overline{\text{WR}}$)	(A12)P2.4	25	Port 2 Bit 4	(Address 12)
Port 3 Bit 7 (Read Strobe)	17	P3.7($\overline{\text{RD}}$)	(A11)P2.3	24	Port 2 Bit 3	(Address 11)
Crystal Input 2	18	XTAL2	(A10)P2.2	23	Port 2 Bit 2	(Address 10)
Crystal Input 1	19	XTAL1	(A9)P2.1	22	Port 2 Bit 1	(Address 9)
Ground	20	Vss	(A8)P2.0	21	Port 2 Bit 0	(Address 8)

Gambar 2.5.¹⁾
Konfigurasi Pin 8751H

3 adalah sebagai berikut :

- RD (P3.7) : Strobe pembacaan Memori Data Luar.

¹⁾ , Embedded controler Handbook Vol I, Intel Co, 1988, hal 10-17

- WR (P3.6) : Strobe penulisan Memori Data Luar.
- T1 (P3.5) : Masukan dari Pewaktu/Pencacah 1.
- T0 (P3.4) : Masukan dari Pewaktu/Pencacah 0.
- INT1 (P3.3): Masukan Interrupt 1.
- INT0 (P3.2): Masukan Interrupt 0.
- TXD (P3.1) : Keluaran pengiriman data untuk Serial Port (asynchronous) atau sebagai keluaran clock (synchronous).
- RXD (P3.0) : Masukan penerima data Serial (asynchronous) atau sebagai masukan/keluaran data (synchronous).
- RST/VPD

Perubahan taraf tegangan dari rendah ke tinggi akan mereset 8751H. Bila tegangan pada pin ini tetap pada taraf tinggi maka saat tegangan VCC turun ke taraf rendah, VPD akan memberikan tegangan catu untuk RAM dalam.

- ALE/ $\overline{P\overline{R}\overline{O}\overline{G}}$ (Address Latch Enable)

Address Latch Enable merupakan sinyal yang digunakan untuk membedakan Port 0 sebagai bus data atau sebagai bus alamat. Apabila ALE berlogika '1', berarti Port 0 berfungsi sebagai bus alamat dan akan berfungsi sebagai bus data jika ALE berlogika '0'. Selain itu pin ini juga menerima pulsa input pada saat pemrograman EPROM.

- $\overline{P\overline{S}\overline{E}\overline{N}}$ (Program Store Enable)

Akan berlogika '0' apabila CPU sedang mengakses memori program eksternal.

- XTAL 1

Masukan ke penguat osilator berpenguatan tinggi. Kaki ini dihubungkan dengan kristal atau sumber osilator dari luar.

- XTAL 2

Keluaran dari penguatan osilator. Kaki ini dihubungkan dengan kristal bila menggunakan sumber osilator dari dalam.

- \bar{EA} /VDD

External Access pada pin ini jika diberi level tegangan high sesuai untuk jenis TTL, maka 8751H akan mengeksekusi instruksi dari ROM/EPROM Internal selama Program Counter belum melebihi 4096. Jika diberi level tegangan low, maka 8751 akan mengambil instruksi dari Memori program tambahan (External). Pada saat melakukan EPROM Programming pin ini diberi level tegangan sebesar +21 volt.

II.3.3 HUBUNGAN KERJA ANTAR SUBSISTEM

Arsitektur Intel 8751H dan hubungan kerja dari subsistem dalam single chip microcomputer Intel 8751H dapat dilihat pada Arsitektur diagram blok (gambar 2.6.), yang menunjukkan bagian-bagian fungsional subsistem yaitu :

- Resident ROM atau ROM Internal.
- Resident RAM atau RAM Internal.
- ALU (Arithmetic Logic Unit).

- Bagian kontrol timing.
- Bagian Special Function Register.
- Bagian Register Umum.
- Bagian Akumulator atau Accumulator Latch.
- Bagian Latch atau Penyangga Port.
- Bagian Program Counter.
- Bagian Program Status Word.
- Bagian RAM Address Register.
- Bagian timer / counter.
- Bagian Port.
- Bagian Oscillator.
- Bagian Power Supply.

Semua subsistem / bagian fungsional dari Intel 8751H ini terangkai menjadi satu hubungan kerja efektif melalui bus internal 8 bit.

II.3.4.CENTRAL PROCESSING UNIT (CPU)

Proses yang dilakukan oleh Microcomputer system dipusatkan pada CPU. Pada minimum sistem dari microcomputer CPU ini berupa microprocessor yang mampu membuat keputusan sesuai dengan program dari Operating System.

□ Program Counter (PC)

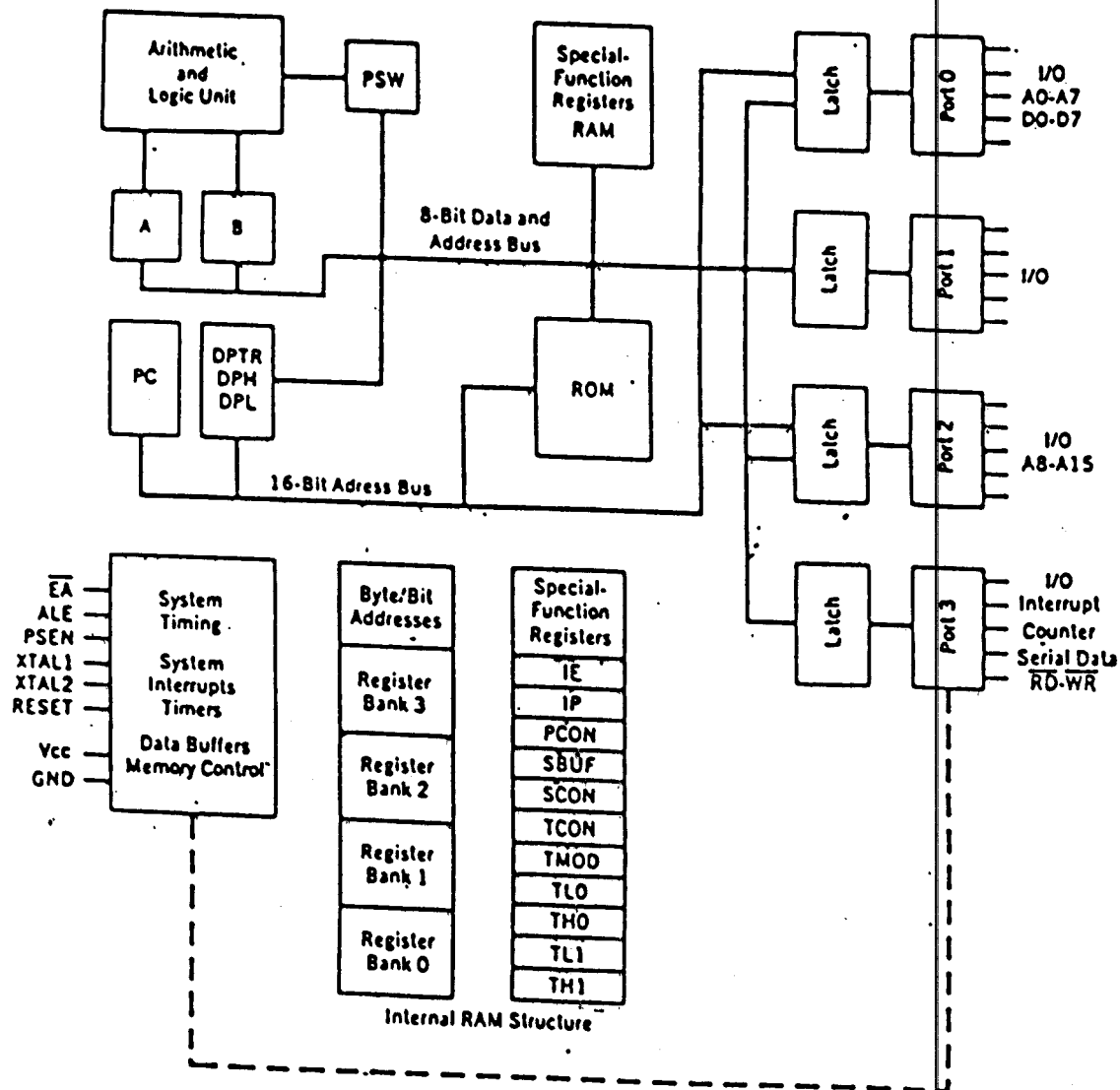
PC merupakan register 16 bit yang digunakan untuk mengontrol urutan instruksi yang akan dijalankan.

□ Dekoder Instruksi

Bagian menerjemahkan setiap instruksi dan membangkitkan sinyal yang akan mengontrol fungsi dari setiap bagian di dalam CPU.

□ RAM Data Dalam

RAM Data Dalam sebesar 128 byte ini terdiri dari :



Gambar 2.6.²⁾
Arsitektur Diagram Blok Keluarga MCS-51

- 4 Bank Register

Setiap Bank terdiri dari 8 register R0 sampai R7.

- 128 bit yang dapat dialamati secara langsung.

Bagian ini terletak pada alamat 20H sampai 2FH pada RAM Data

Dalam.

□ Stack

Kedalaman stack dibatasi oleh RAM Data Dalam.

□ Register Fungsi Khusus (SFR)

Register-register dalam SFR adalah :

□ Akumulator (ACC) atau Register A

□ Register B

Digunakan bersama register A untuk instruksi perkalian dan pembagian.

□ Program Status Word (PSW)

Register ini meliputi bit-bit : CY (Carry), AC (Auxiliary Carry), F0 (sebagai flag), RS0 dan RS1 (untuk pemilihan register bank), OV (overflow), dan P (Parity flag).

□ Penunjuk Stack (Stack Pointer/SP)

SP adalah register 8 bit yang menunjuk alamat dari byte terakhir yang dimasukkan ke stack, juga sebagai alamat dari byte selanjutnya yang akan dikeluarkan.

□ Penunjuk Data Tinggi (Data Pointer High/DPH) dan

Penunjuk Data Rendah (Data Pointer Low/DPL)

DPTR 16 bit adalah gabungan dari DPH dan DPL yang masing-masing 8 bit.

DPTR digunakan pengalamatan register tak langsung untuk pemindahan konstanta Memori Program, pemindahan data dengan Memori Data Luar dan untuk bercabang sampai 64 Kbyte.

- Port 0, Port1, Port 2, Port 3

Semua Port dapat dialamati secara byte maupun bit.

- Register Prioritas Interrupt (Interrupt Priority

Register/IP) Berisi bit-bit kontrol untuk mengaktifkan interrupt pada taraf yang diinginkan.

- Interrupt Enable Register (IE)

IE berisi bit-bit untuk mematikan/mengidupkan setiap sumber interrupt atau keseluruhan.

- Timer/Counter Mode Register (TMOD)

Bit-bit pada register TMOD digunakan untuk memilih pewaktu atau pencacah yang bekerja.

- Timer/Counter Control Register (TCON)

Bit-bit pada TCON digunakan untuk memulai/menghentikan pewaktu pencacah, flag-flag overflow dan permintaan interrupt.

- TH0, TL0, TH1 dan TL1

TH0 dan TH1 digunakan untuk byte tinggi dari pewaktu, pencacah 0 dan 1. TL0 dan TL1 untuk byte rendahnya. Register-register ini dapat dibaca dan ditulis.

- Serial Control Register (SCON)

Bit-bit pada SCON digunakan untuk pemilihan mode operasi dari

serial port.

□ Serial Data Buffer (SBUF)

SBUF digunakan untuk menampung data masukan atau keluaran dari serial port tergantung apakah serial port menerima atau mengirim data.

□ Bagian Aritmetika

Bagian aritmetika dari prosesor membentuk beberapa fungsi manipulasi data yang dilaksanakan olehh Aritmetic/Logic Unit, register A, B dan PSW.

□ Rangkaian Osilator

Rangkaian Osilator yang terdapat di dalam serpihh adalah rangkaian paralel anti-resonant dengan batas frekuensi mulai dari 1,2 MHz samapai 12MHz.

□ Prosesor Boolean

Prosesor Boolean adalah prosesor bit yang berdiri sendiri, yang memiliki sendiri pasangan instruksi, akumulator dan bit addressable RAM dan I/I. Prosesor Boolean dapat juga membentuk operasi-operasi bit seperti set, clear, complement, jump-if-set, jump-if-not-set, jump-if-set-then-clear dan pemindahan dari atau ke carry.

II.3.5 MEMORI

Pada microcontroler MCS-51, program data memori memiliki alamat yang berbeda. Perbedaan alamat ini memungkinkan Data Memory di-akses melalui alamat 8-bit, yang lebih mudah disimpan dan dimanipulasi olehh CPU 8 bit. Walaupun begitu Alamat Data Memory yang terdiri dari 16 bit juga dapat dilakukan melalui register DPTR. Program Memory hanya dapat dibaca, alamat yang tersedia samapai

64 Kbyte. 4 Kbyte awal tersedia didalam chip (on-chip). Data Memory menempati alamat yang berbeda dengan Program Memory. Ram Eksternal dapat mencapai 64 Kbyte. RAM Internal terdiri dari 256 byte yang terbagi 2 bagian yaitu 128 byte terendah bagi Internal RAM yang dapat dialamati baik scara Direct maupun Indirect Addressing. 128 byte berikutnya digunakan sebagai SFR (Special Function Register) yang hanya dialamati secara langsung (Direct Addressing).

II.3.5.1. Read Only Memory (ROM)

ROM merupakan memori yang hanya dapat dibaca. Informasi yang dapat dibaca dari ROM dapat berupa instruksi atau data.

Memori Program residen terdiri atas 4 Kbyte dengan masing-masing word terdiri dari 8 bit, yang dialamati oleh program counter. Pada Intel 8751H yang digunakan adalah EPROM (Erasable Programmable ROM).

Pemetaan alokasi ROM dapat dilihat pada gambar 2.7.

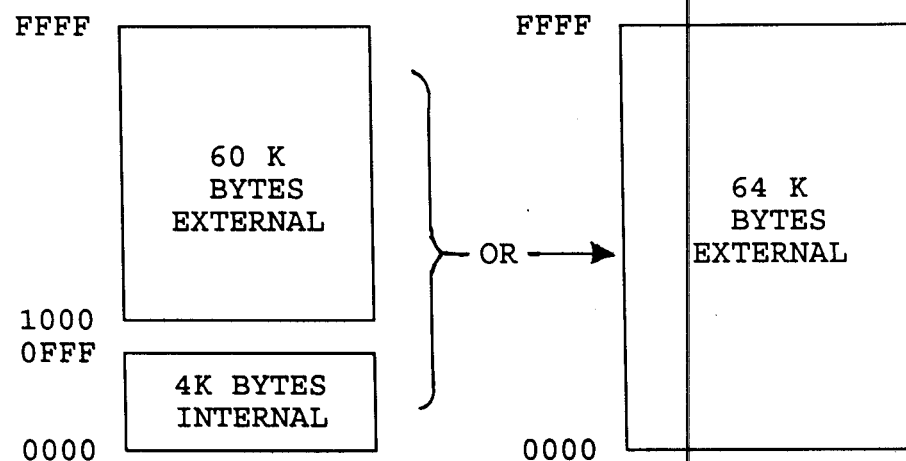
Memori program internal Intel 8751H dapat diisi dengan mengaktifkan pin-pin yang berhubungan dengan pemrograman EPROM. Pin yang perlu diaktifkan adalah :

1. Xtal, pin ini berfungsi sebagai clock output, frekuensi kristal yang digunakan 4 - 6 MHz.
2. RESET : pin reset digunakan untuk proses inisialisasi.
3. EA/Vpp : sebagai pin untuk mengaktifkan proses pemrograman EPROM.
4. Port 0 : 8 bit bus yang digunakan sebagai bus data dan bus alamat.
5. Port 1 : menyatakan address dari lokasi EPROM yang akan diisi (address 0 - 7).

6. Port 2 : Port 2 bit 0-3 digunakan sebagai address 8-11,
 Port 2 bit 4-5 tidak digunakan,
 Port 2 bit 6 diberi logic 'low',
 Port 2 bit 7 diberi logic 'high'.
7. VPP : sebagai pin power supply pemrograman.
8. $\overline{P\overline{R}\overline{O}\overline{G}}$: sebagai pulsa input program.

II.3.5.2. Random Access Memory (RAM)

RAM Internal merupakan Data Memory yang terdiri atas 128 bytes pada address 00H - 7FH serta address 80H - FFH yang digunakan oleh 21 buah register khusus yang disebut SFR (Special Function Register). Ke-128 bit awal RAM ditunjukkan pada gambar 2.8.



Gambar 2.7.
Peta Alokasi ROM

Dimana 32 byte terendah dikelompokkan menjadi 4 buah bank yang terdiri atas 8 buah register dengan alamat 00H - 1FH. Instruksi pada Program mengenal register-register ini sebagai R0 - R7. Dua bit pada Program Status Word (PSW) akan memilih register Bank mana yang akan digunakan yaitu PSW 3 dan PSW 4. Dengan cara ini penggunaan kapasitas memori akan lebih efisien, karena instruksi register lebih pendek dibandingkan dengan instruksi yang menggunakan pengalamatan langsung. 16 byte berikutnya (20H - 2FH) yang terletak diatas register Bank membentuk suatu lokasi memori yang mampu dialamati secara bit (bit addressable) dan dapat digunakan sebagai general purpose register.

II.3.6. PEWAKTU/PENCACAH

register-register yang digunakan sebagai pewaktu/pencacah adalah TH1, TL1 (pewaktu/pencacah 1), dan TH0, TL0 (pewaktu/pencacah 0).

Terdapat 4 mode operasi dari Pewaktu/Pencacah, yaitu:

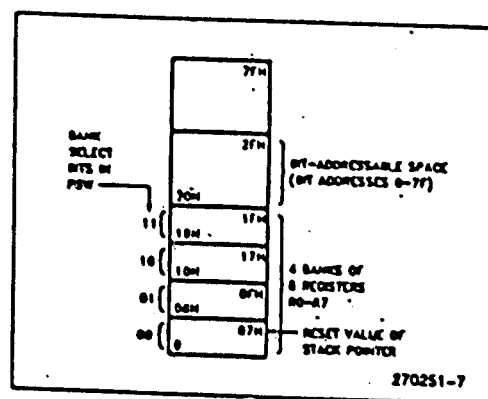
- Mode 0 (Pewaktu/Pencacah 8 bit dengan 32 prescaler) TH0 atau TH1 adalah pewaktu/pencacah. Sedang register TL0 atau TL1 bit 4-0 sebagai prescaler dan bit 7-5 tidak dipakai.
- Mode 1 (Pewaktu/Pencacah 16 bit) TH1 dan TL1 atau TH0 dan TL0 dikaskade.
- Mode 2 (Pewaktu/Pencacah 8 bit autoreload) TH1 atau TH0 berisi bilangan yang akan diisikan ke TL1 atau TL0 setiap kali overflow.
- Mode 3

Pewaktu 0 dibagi menjadi dua buah pewaktu/pencacah 8 bit yang terpisah.

Pewaktu 1 tidak berjalan.

Fungsi bit-bit pada register TMOD (Timer/Counter Mode Register) yang diperlihatkan pada tabel 2.2. adalah sebagai berikut :

- Gate : Bila Gate = 1, Pewaktu/Pencacah 'x' enable hanya saat pin INTx tinggi dan TRx = 1. Saat Gate = 0, Pewaktu/Pencacah enable jika bit TRx = 1.
- C/T : Jika bit C/T = 0, maka Pewaktu/Pencacah 'x' akan berfungsi sebagai pewaktu. Jika C/T = 1, maka Pewaktu/Pencacah 'x' beroperasi sebagai pencacah.
- M1, M0 : (0,0) untuk Mode 0, (0,1) untuk mode 1, (1,0) untuk mode 2 dan (1,1) untuk mode 3.



Gambar 2.8.
Peta Alokasi RAM



³⁾Tabel 2.2.Register TMOD

Gate	C/T	M1	M0	Gate	C/T	M1	M0
------	-----	----	----	------	-----	----	----

(MSB)

(LSB)

II.3.7. SISTEM INTERRUPT

MCS-51 mempunyai 5 buah sumber interrupt yang dapat membangkitkan interrupt request, yaitu :

- INT0 : Permintaan interrupt luar dari kaki P3.2.
- INT1 : Permintaan interrupt luar dari kaki P3.3.
- Pewaktu/Pencacah 0 (bila terjadi overflow).
- Pewaktu/Pencacah 1 (bila terjadi overflow).
- Port Serial (bila pengiriman/penerimaan satu frame telah lengkap).

Tiap sumber interrupt mempunyai alamat tertentu, yang diawali pada 0003H dan seterusnya dengan selang 8 byte. Alamat awal dari tiap-tiap interrupt request ini terdapat pada Tabel 2.3. Saat terjadi interrupt CPU secara otomatis akan menuju ke subroutine pada alamat tersebut. Setelah interrupt service routine selesai dikerjakan, CPU akan mengerjakan program semula.

⁴⁾Tabel 2.3.Alamat Awal dari Interrupt Service Program

Sumber Interrupt	Alamat awal
Interrupt Luar 0 (INT0)	3 (0003 H)

³⁾ Ibid, hlm.6-7

⁴⁾ Ibid, hhlm 6-22

Pewaktu/Pencacah 0 (T0)	11 (000B H)
Interrupt Luar 1 (INT1)	19 (0013 H)
Pewaktu/Pencacah 1 (T1)	27 (001B H)
Port Serial	35 (0023 H)

Sumber interrupt dapat dibuat enable/disable dengan mengatur bit-bit pada register IE (Interrupt Enable). Bit-bit pada register IE adalah sebagai berikut :

- EA (IE.7) : Bila EA = 0, maka semua interrupt akan dimatikan dan tidak terpengaruh status dari IE.4 - IE.0.
- ES (IE.4) : untuk interrupt port serial.
- ET1 (IE.3) : untuk interrupt pewaktu/pencacah 1.
- EX1 (IE.2) : untuk interrupt luar 1.
- ET0 (IE.1) : untuk interrupt pewaktu/pencacah 0.
- EX0 (IE.0) : untuk interrupt luar 0.

Pengaturan prioritas dari interrupt request terdapat pada register IP (Interrupt Priority). Bit-bit pada register IP adalah sebagai berikut :

- PS (IP.4) : untuk prioritas serial port.
- PT1 (IP.3) : untuk prioritas pewaktu/pencacah 1.
- PX1 (IP.2) : untuk prioritas interrupt luar 1.
- PT0 (IP.1) : untuk prioritas pewaktu/pencacah 0.
- PX0 (IP.0) : untuk prioritas interrupt luar 0.

Fungsi dari bit-bit pada register TCON (Timer/Counter Control Register) adalah sebagai berikut :

- TF1 (TCON.7) dan TF0 (TCON.5) :

TFx = 1 saat pewaktu/pencacah 'x' overflow, dan TFx = 0 ketika interrupt dilaksanakan.

- TR1 (TCON.6) dan TR0 (TCON.4) :

Jika TRx diisi '1' atau '0' maka pewaktu/pencacah 'x' akan aktif atau berhenti.

- IE1 (TCON.3) dan IE0 (TCON.1) :

Bit IEx = 1 bila sisi jatuh dari interrupt luar 'x' terdeteksi dan akan diisi '0' oleh perangkat keras bila interrupt dilayani.

- IT1 (TCON.2) dan IT0 (TCON.0) :

Bila ITx = '1' atau '0' maka interrupt luar 'x' akan bekerja saat sisi jatuh atau taraf rendah.

II.3.8 PORT SERIAL

Port Serial dalam serpih MCS-51 ini dapat dioperasikan dalam 4 mode, yaitu:

- Mode 0 (synchronous)

Data serial 8 bit dikirim dan diterima melalui RXD, dengan bit terendah (LSB) yang pertama dan TXD mengeluarkan clock penggeser. Baud rate untuk mode 0 adalah 1/12 dari frekuensi osilator.

- Mode 1 (asynchronous)

TXD mengirim atau RXD menerima 10 bit dengan baud rate dapat diubah-ubah.

- Mode 2 (asynchronous)

TXD mengirim atau RXD menerima 11 bit dengan baud rate adalah 1/32 atau 1/64 dari frekuensi osilator.

- Mode 3 (asynchronous)

Mode 3 ini sama dengan mode 2 kecuali pada mode 3 baud rate dapat diubah-ubah.

Register SCON digunakan untuk mendefinisikan mode operasi dari kontrol fungsi-fungsi dari port serial. Register SCON digambarkan pada tabel 2.4.

Fungsi bit-bit pada register SCON tersebut adalah :

⁵⁾Tabel 2.4.
Register SCON

SM0	SM1	SM2	REN	TB8	RB8	T1	R1
-----	-----	-----	-----	-----	-----	----	----

(MSB)

(LSB)

- SM0, SM1 : pemilihan Mode (0,0) untuk Mode 0, (0,1) untuk Mode 1, (1,0) untuk Mode 2 dan (1,1) untuk Mode 3.
- SM2 : Pada Mode 2 dan 3, bila SM2 = 1, maka RI tidak akan diaktifkan bila bit data ke 9 (RB8) yang diterima adalah '0'. Pada Mode 1, jika SM2 = 1, maka RI tidak akan diaktifkan bila stop bit yang diterima tidak sempurna.
- REN : Bit enable penerima data.
- TB8 : Bit data ke 9 yang akan dikirim pada Mode 2 dan 3.
- RB8 : Pada Mode 2 dan 3, RB8 sebagai bit data ke 9 yang diterima. Pada Mode 1, bila SM2 = 0, RB8 adalah stop bit yang diterima.

⁵⁾ Ibid, hlm.39

Pada Mode 0, RB8 tidak digunakan.

- TI : Transmit interrupt flag diisi '1' oleh perangkat keras pada akhir dari pengiriman.
- RI : Receive interrupt flag diisi '1' oleh perangkat keras pada akhir dari penerimaan. Semua data yang dikirim maupun diterima ditempatkan pada Serial Data Buffer Register (SBUF).

II.4 KOMUNIKASI DATA SERIAL

Komunikasi data serial adalah dengan mengirimkan satu bit data setiap selang waktu tertentu sehingga praktis kecepatannya relatif lebih lambat dibanding dengan komunikasi data paralel. Salah satu serial interface yang sering digunakan adalah RS-232. RS-232 adalah interface Standart Electronic Industry Association (EIA). Prinsip kerja dari RS-232 adalah mengubah data paralel menjadi data serial atau sebaliknya untuk kemudian dikirimkan dalam bentuk data biner serial.

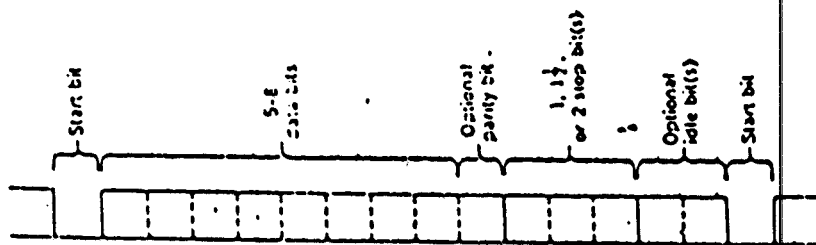
Metode pengiriman sinyal yang dipakai RS-232 adalah metode asinkron. Komputer walaupun menggunakan transmisi asinkron untuk komunikasi serial, data yang diterima akan diubah menjadi sinkron. Transmisi asinkron menyusun karakter yang ditransmisikan antara satu start bit dan dua stop bit seperti yang ditunjukkan pada gambar 2.9. dibawah ini.

Kecepatan transmisi dalam saluran biasanya diekspresikan dalam baud rate, sebenarnya istilah baud menunjukkan jumlah sinyal yang ditransmisikan dinyatakan dalam keadaan bit. Istilah baud sama dengan bit tiap detik. Agar tidak menimbulkan kerancuan dipakai

istilah baud untuk menunjukkan kecepatan transmisi, sedangkan untuk kecepatan informasi dipakai bit per detik (bps).

II.4.1 KONEKTOR DB-25

Konektor yang dipakai untuk menggunakan RS-232C dengan media transmisi adalah konektor DB dengan 25 pin, biasanya dikenal dengan sebutan DB-25. Berdasarkan dari bentuknya DB-25 mempunyai dua jenis yaitu jenis male dan female. Pada IBM Asynchronous Communication Adapter yang sesuai dengan standart EIA RS-232 memakai konektor male. Ukuran mekanis dari konektor ini sudah berdasarkan ukuran standard ISO. Gambar 2.10. menjelaskan karakter mekanis dan posisi pin-pinya.



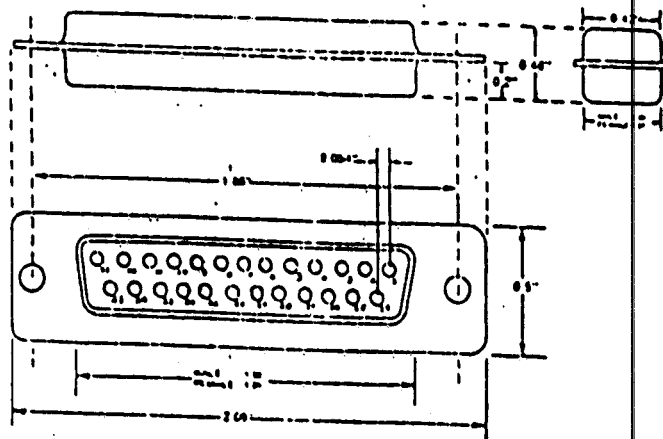
Gambar 2.9.
Transmisi Asinkron

II.4.2 FORMAT DATA

Seperti halnya transmisi asinkron pada umumnya, format data RS-232 diawali dengan start bit, kemudian data bit dan diakhiri dengan parity dan stop bit.

Start Bit

Pada sistem komputer atau suatu peralatan penerima karakter asinkron yang dapat dibaca baik logika '0' atau logika '1' pada jalur received data akan memberikan kondisi awal logika '0' selama bit time yang sudah ditentukan, bila data word akan dikirimkan. Dengan adanya perubahan kondisi '1' ke '0', komputer akan menafsirkan bahwa suatu word data akan dikirim.



⁶⁾Gambar 2.10.
Karakteristik konektor DB-25

⁶⁾Hall, Microprocessor and Interfacing, hhal. 450

Bit Data

Proses transmisi harus mengikuti beberapa aturan. Salah satu yang harus diperhatikan adalah kode karakter yang dipakai dalam bit data. Saat ini ada beberapa kode yang cukup populer antara lain : kode 5 bit untuk Baudot Murray; kode 6 bit untuk IBM coorespondence; kode 8 bit untuk EBCDIC; dan yang paling populer adalah kode 7 bit untuk kode ASCII. Standard RS-232 tidak membatasi jumlah bit per karakter, akan tetapi komputer membatasinya. Pengiriman data bit selalu dimulai dengan bit terkecil sampai bit terbesar.

Parity dan Stop Bit

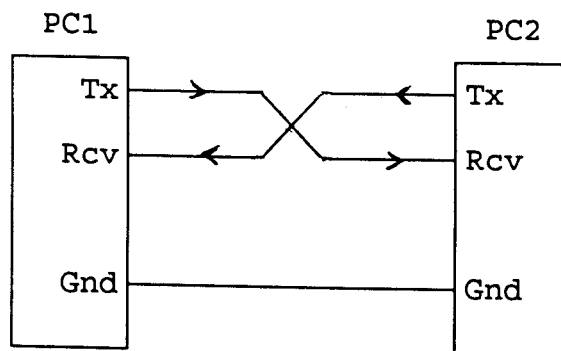
Setelah data bit akan diikuti dengan parity bit, yang digunakan untuk mendeteksi kesalahan. Suatu kesalahan akan terjadi bilamana ada gangguan pada pulsa yang dikirim, oleh karenan itu satu bit perlu ditambahkan untuk mengecek kesalahan dalam pembacaan data dan biasanya disebut parity cek.

Bit terakhir yang ditransmisikan adalah stop bit. Bit ini boleh dikatakan bukan bit yang sebenarnya, tetapi stop times yang mengizinkan received data untuk mengumpulkan bit-bit serial yang sudah diterima dan siap untuk karakter selanjutnya.

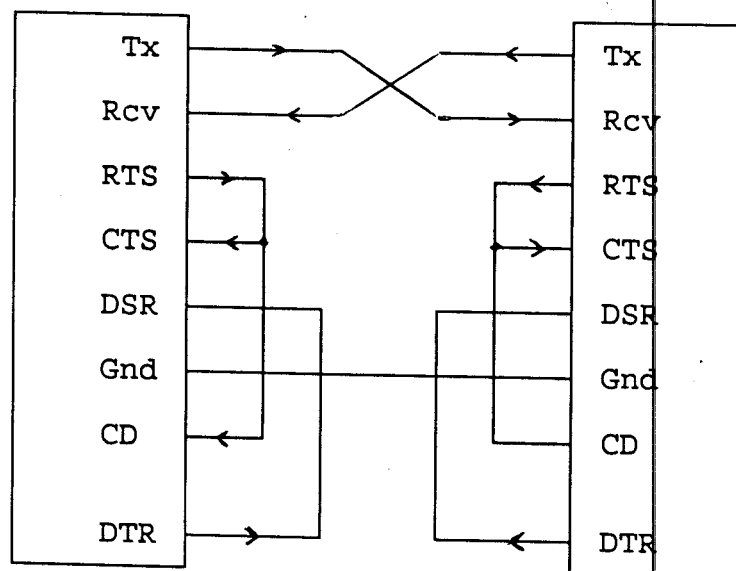
II.4.3 INTERKONEKSI KABEL

Komunikasi langsung antara terminal data dengan komputer yang melalui RS-232C yang paling mudah ditunjukkan pada Gambar 2.11. Pin 2 (TxD) dari tereminal

data 1 dihubungkan dengan pin 3 (TxD) dari PC dan sebaliknya. Tidak satupun dari pin-pin lainnya kecuali pin 7 (Groubd), yang merupakan common return untuk rangkaian penerima atau pengirim.



a. Simple null modem connection



b. General-purpose null modem connection

⁷⁾Gambar 2.11.

Hubungan Antara PC dan PC

⁷⁾Kruglinski, David, Ouide to IBM PC Communication, The Orsbone/ McGraw-Hill, 1986, hal.179

II.5 MOTOR STEPPER

Motor stepper termasuk motor sinkron, dimana terdapat hubungan antara perintah masukan (input) dengan perubahan posisi yang dihasilkan. Input motor stepper ini dapat dihubungkan dengan sistem digital secara langsung dan dapat berputar searah ataupun berlawanan jarum jam.

Pengoperasian motor stepper didasarkan pada prinsip dasar magnetik. dimana kutub magnet yang berlawanan tarik-menarik dan kutub magnet yang sama akan tolak-

menolak. Prinsip kerja dari motor stepper ini diperlihatkan seperti pada gambar 2.12. Jika kumparan stator diberi aliran listrik sedemikian sehingga stator A menjadi kutub utara dan stator B menjadi kutub selatan, dan rotor (permanan magnet) pada posisi seperti pada gambar 2.12.a., maka torsi yang dibangkitkan akan memutor 180 drejat dari posisi semula. Tetapi kenyataannya hal tersaebut tidak akan menjadi dan rotor tidak akan bergerak karena gaya yang terjadi dalam keadaan seimbang (balance).

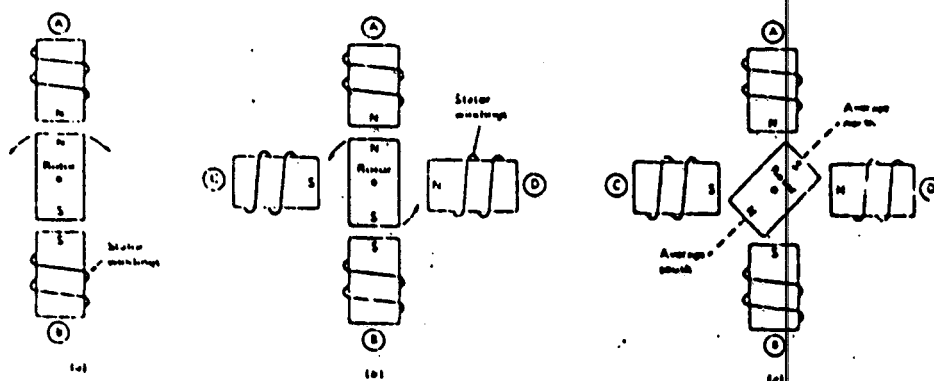
Jika ditambahkan dua kutub stator yaitu kutub stator C dan D seperti pada gambar 2.12.b. dan kemudian diberi energi maka dapat ditentukan arah putaran dari rotor. Pada Gambar 2.12.b. arah putaran berlawanan dengan arah jarum jam dan rotor akan berhenti bergerak setelah kutub selatan permanen magnet terletak antara ke dua kutub utara stator dan begitu pula sebaliknya seperti pada gambar 2.12.c.

Motor stepper dapat dibagi menjadi 3 jenis bila ditinjau dari konstruksi dan kemampuannya :

II.5.1 Variable Reluctance Motor

Motor Variable Reluctance (VR) mempunyai sebuah stator dengan sejumlah kutub yang terbuat dari kumparan. Rotor berbentuk silinder dengan beberapa unit gigi yang mempunyai kaitan dengan kutub pada stator. Jumlah gigi pada stator berkaitan dengan sudut gerak setiap langkah, seperti terlihat pada gambar 2.13.

Ketika arus mengalir melalui kumparan, torsi akan dibangkitkan, sejalan dengan itu rotor akan berputar ke posisi reluktansi magnetik berharga minimum. Posisi ini merupakan posisi yang stabil dan masih banyak posisi stabil yang dapat dicapai sesuai dengan pola mengenergian pada stator. Ketika kumparan diberi energi, medan stator



^{a)}Gambar 2.12.
Prinsip dasar Putaran Motor Stepper

^{a)}Charles A.Schuller, William L.Mc.Name, Industrial Electronics and Robotics, McGraw Hill Inc, 1986, hlm 53

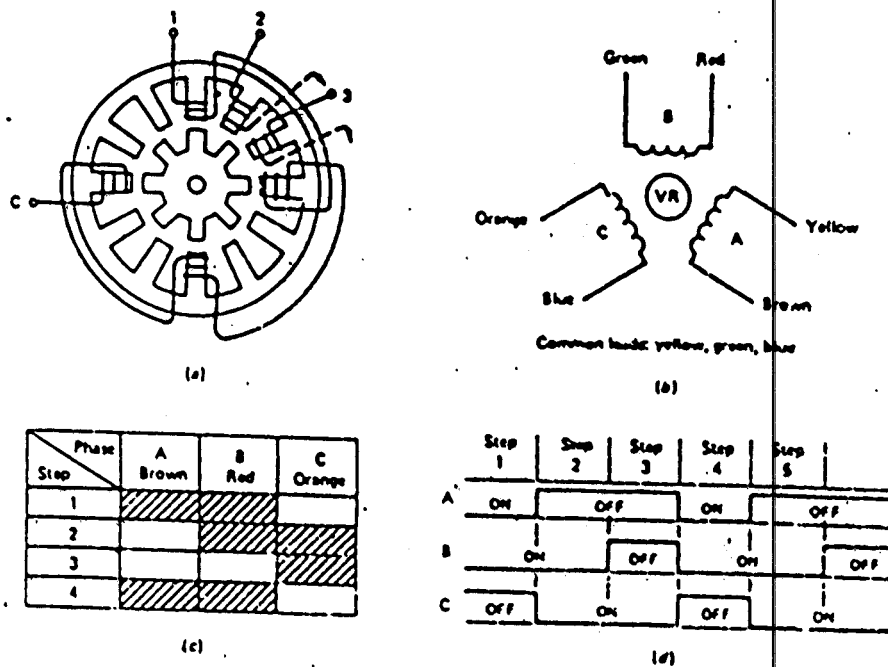
berubah menyebabkan rotor bergerak pada posisi stabil yang baru. Gambar 2.13.c. menggambarkan model dasar pengeksitasian yang akan menghasilkan sudut gerak yang cukup baik. Pengeksitasian dengan dua kumparan ON merupakan pilihan yang tepat karena akan menghasilkan torsi yang cukup besar.

Motor stepper jenis variable reluctance mempunyai sisa magnetik yang sangat kecil sehingga tidak akan ada gaya yang bekerja pada rotor, ketika stator tidak diberi energi. Sudut langkah dari motor jenis ini relatif kecil dan juga memiliki kapasitas kelembaman yang kecil dibandingkan dengan yang lainnya.

II.5.2. Permanent Magnet Motor

Ciri utama dari motor jenis ini, selalu terdapat magnet permanent pada konstruksi rotornya. Untuk menghasilkan gerakan atau langkah yang lebih baik, perlu ditambahkan beberapa kutub pada stator dan gigi-gigi pada bagian stator demikian juga pada rotornya. Jumlah gigi-gigi pada stator dan rotor menentukan sudut langkah yang akan dicapai pada saat salah satu kutub dari kumparan berubah.

Motor jenis ini beroperasi berdasarkan interaksi fluks magnet pada rotor dan gaya magnet pada stator. Jika kumparan dieksitasi dengan pola yang telah ditentukan, rotor akan bergerak pada titik keseimbangan yang baru dan berputar sesuai dengan perubahan pola seperti pada gambar 2.14. Jika kumparan rotor tidak dieksitasi maka tidak akan ada torsi penahan pada rotor.



⁹⁾Gambar 2.13.
Variable Reluctance Motor

- Bentuk kumparan VR motor
- Hubungan kode warna dan phase setiap kumparan
- Bentuk dual-excitation tiga phase
- Bentuk gelombang

⁹⁾Ibid, hlm 54

Step	Switch #1	Switch #2
1	1	5
2	1	4
3	3	4
4	3	5
5	1	5

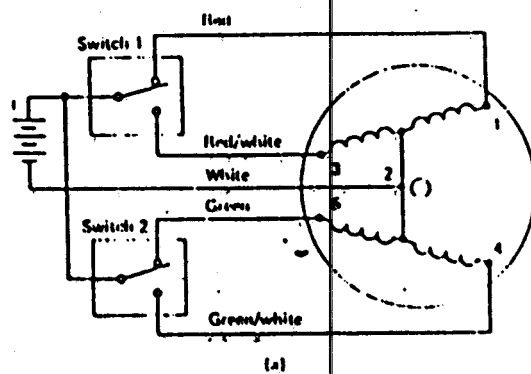
* To reverse direction, read chart up from bottom

(b)

Phase	FWD →				
	1 Step	2 Step	3 Step	4 Step	5 Step
1	ON	ON	OFF	OFF	ON
3	OFF	OFF	ON	ON	OFF
5	ON	OFF	OFF	ON	ON
4	OFF	ON	ON	OFF	OFF

← REV

(c)



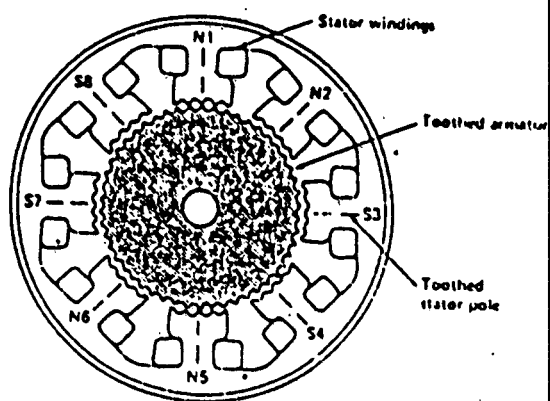
¹⁰⁾Gambar 2.14.
Permanent Magnet Motor

- Diagram skema
- Deretan pulsa
- Bentuk Gelombang

¹⁰⁾Ibid, hal.53

II.5.2 Permanent Magnet Hybrid Motor

Konstruksi Permanent Magnet Hybrid Motor terdiri dari rotor yang merupakan kombinasi antara motor jenis Variable Reluctance dan Permanent Magnet. Konstruksi dari motor ini seperti pada gambar 2.15. Stator terdiri atas kumparan stator dan rotornya memakai gigi-gigi. Dengan konstruksi seperti ini motor mempunyai kapasitas torsi yang cukup besar dengan ketelitian langkah sekitar 3 persen dan sudut langkah 0,5 - 15 derajat. Motor ini mempunyai kecepatan 1000 langkah per menit bahkan lebih besar.



¹¹⁾Gambar 2.15.
Permanent Magnet Hybrid Motor

¹¹⁾Ibid, hlm.54

BAB III

PERENCANAAN PERANGKAT KERAS

III.1. PENDAHULUAN

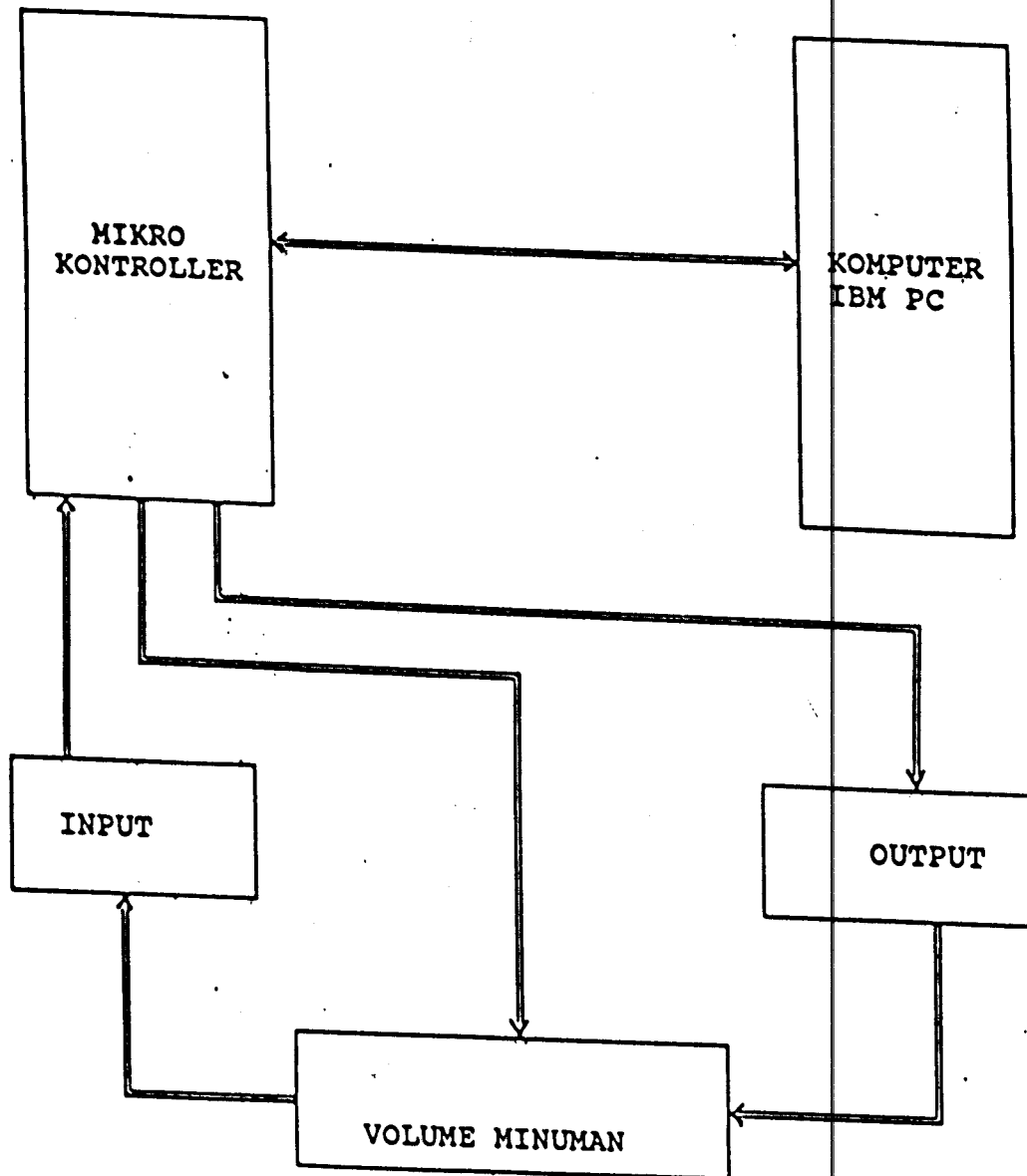
Alat yang direncanakan mempunyai jumlah masukan 24 titik, keluaran 24 titik, dan rangkaian serial. Rangkaian pengkondisi sinyal yang dibuat terdiri dari 8 buah rangkaian untuk keluaran dan masukan 12 volt. Dan 8 buah rangkaian untuk keluaran dan masukan 5 volt.

III.2. DIAGRAM BLOK

Diagram blok dari Alat yang direncanakan ditunjukkan pada Gambar 3.1.

IBM PC - AT digunakan untuk mengirimkan data volume minuman ke mikrokontroller, dimana pengiriman data ini dilakukan secara serial karena pada mikrokontrooler sendiri ada fasilitas untuk hubungan serial. Data ini nantinya disimpan di RAM internal dari mikrokontroller.

Unit masukan/keluaran merupakan bagian dari sistem yang berhubungan langsung dengan masukan/keluaran.

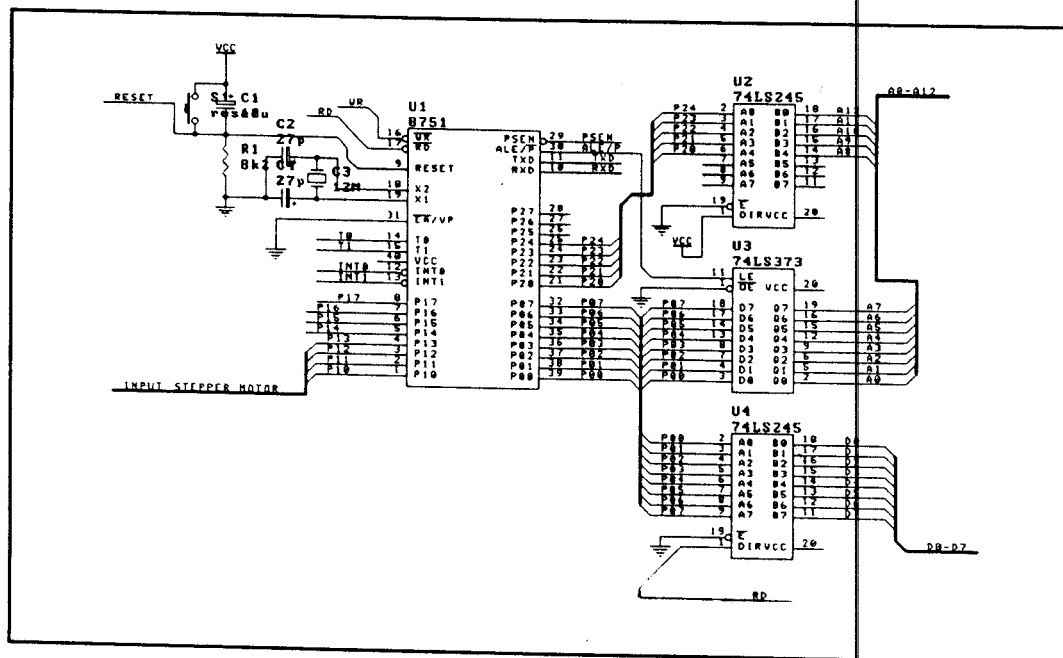


Gambar 3.1.

DIAGRAM BLOK DARI ALAT YANG DIRENCANAKAN

III.3. RANGKAIAN MIKROKONTROLLER 8751H

Rangkaian mikrokontroller 8751 diperlihatkan pada 3.2. Kaki EA arus
dihubungkan ke Vcc karena mikrokontroller ini memakai eprom yang ada dalam
serpihnya. Rangkaian Latch diperlukan untuk memisahkan alamat rendah dan data
dari port 0. Pada perencanaan ini digunakan 8



Gambar 3.2.

Rangkaian Mikrokontroler 8751H

bit Latch 74LS373. Enable (Kaki 1) dihubungkan dengan kaki ALE dari 8751. Bila sinyal ALE aktif, alamat rendah akan terkunci pada bus alamat.

Pin DIR dari 74LS245 sebagai buffer data akan high, yang berarti buffer mempunyai arah dari A ke B, bila RD tidak aktif. Bila sinyal RD aktif, yaitu 8751H membaca data input dari IC 8255, maka pin DIR akan berlogika rendah

dan arah dari B ke A. Dan saat WR aktif (RD high), arah buffer memungkinkan 8751H dapat menulis ke port output IC 8255.

III.4. RANGKAIAN DEKODER MIKROKONTROLLER 8751H

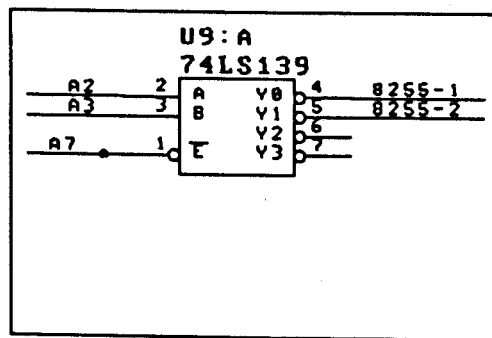
Rangkaian dekoder diperlukan untuk memberikan alamat tertentu pada port masukan dan port keluaran. Pemetaan I/O ditunjukkan pada Tabel 3.1.

Dari tabel kebenaran tersebut terlihat dapat dibuat rangkaian dekoder menggunakan 74LS139 seperti terlihat pada Gambar 3.3.

Tabel 3.1.

PEMETAAN MEMORI DARI I/O

	A7	A6	A5	A4	A3	A2	A1	A0
PortA1	0	0	0	0	0	0	0	0
PortB1	0	0	0	0	0	0	0	1
PortC1	0	0	0	0	0	0	1	0
CW1	0	0	0	0	0	0	1	1
PortA2	0	0	0	0	0	1	0	0
PortB2	0	0	0	0	0	1	0	1
PortC2	0	0	0	0	0	1	1	0
CW2	0	0	0	0	0	1	1	1



Gambar 3.3.

RANGKAIAN DEKODER

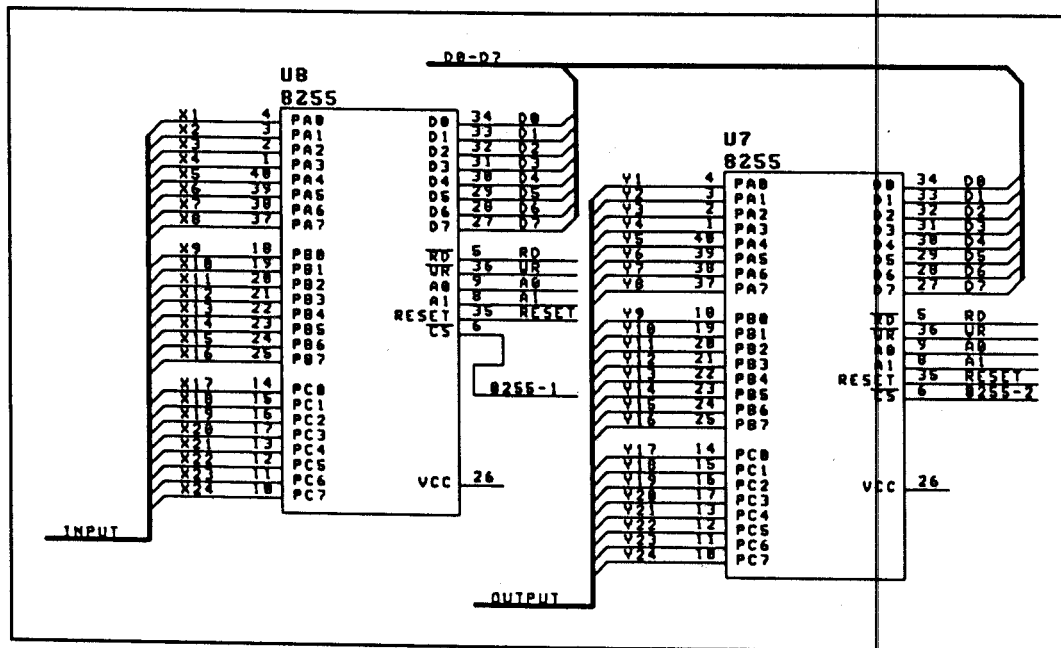
III.5. RANGKAIAN MASUKAN/KELUARAN

PPI 8255 digunakan sebagai port masukan/keluaran. Dalam hal ini 8255 dioperasikan pada Mode 0. PPI 1 digunakan sebagai 24 bit port masukan dan PPI 2 sebagai 24 bit port keluaran.

Pemilihan 8255 yang diaktifkan tergantung pada masukan CS yang dihubungkan dengan keluaran rangkaian dekoder. Sinyal kontrol RD, WR dan RST dari mikrokontroller 8751H langsung dihubungkan dengan kaki-kaki RD, WR dan RESET dari PPI, seperti ditunjukkan oleh Gambar 3.4.

III.6. RANGKAIAN PENGKONDISI SINYAL

Rangkaian Pengkondisi Sinyal yang digunakan baik untuk masukan maupun keluaran. Sinyal masukan dan keluaran dengan taraf TTL diubah ke taraf 12 volt dc. Dan untuk sinyal +12 volt menjadi taraf TTL. Rangkaian ini menggunakan optocoupler sebagai pengisolasi listrik antara sumber dan beban.

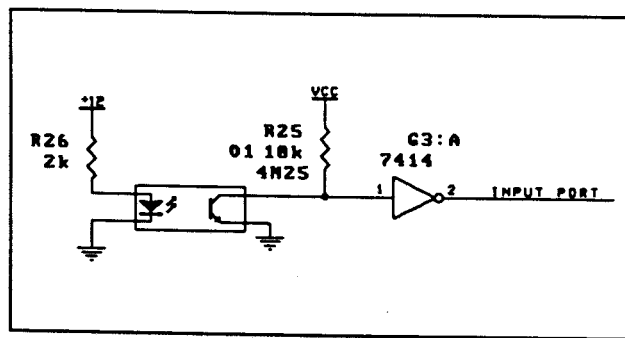


Gambar 3.4.

RANGKAIAN MASUKAN/KELUARAN

Rangkaian pengubah sinyal taraf +12 volt menjadi taraf TTL diperlihatkan pada Gambar 3.5. Current Transfer Ratio dari suatu optocoupler adalah perbandingan dari arus keluaran terhadap arus bias maju LED, yang dapat dinyatakan dengan $N = I_L/I_F$.

Gambar 3.6. menunjukkan karakteristik Current Transfer Optocoupler 4N25 dimana I_L adalah arus kolektor dari phototransistor. Bila harga tahanan yang seri dengan LED dari 4N25 diambil 1 k ohm, sedangkan tegangan bias maju LED dari 4N25 = 1,1 volt, maka arus I_F dapat dihitung :



Gambar 3.5.

RANGKAIAN PENGUBAH SINYAL TARAF 12V KE TTL

$$I_F = (12 - 1,1)/1k$$

$$= 10,9 \text{ mA}$$

Dari karakteristik Current Transfer 4N25, bila $I_F = 10,9 \text{ mA}$ pada suhu kamar maka arus keluaran $I_C = 4 \text{ mA}$. Agar phototransistor bekerja pada daerah non-aktif maka:

$$I_C > (5 - V_{CEsat}) / R_C$$

$$R_C > (5 - 0,2) / 4\text{mA}$$

$$> 1,2 \text{ k ohm}$$

Pada Rangkaian tersebut diambil harga $R_C = (560 + 330 + 2k) = 2,89 \text{ k ohm}$.

Keluaran Phototransistor tersebut dihubungkan ke port masukan melalui Inverter Schmitt Triggered 74LS14, agar diperoleh masukan yang tidak terganggu akibat debounce.

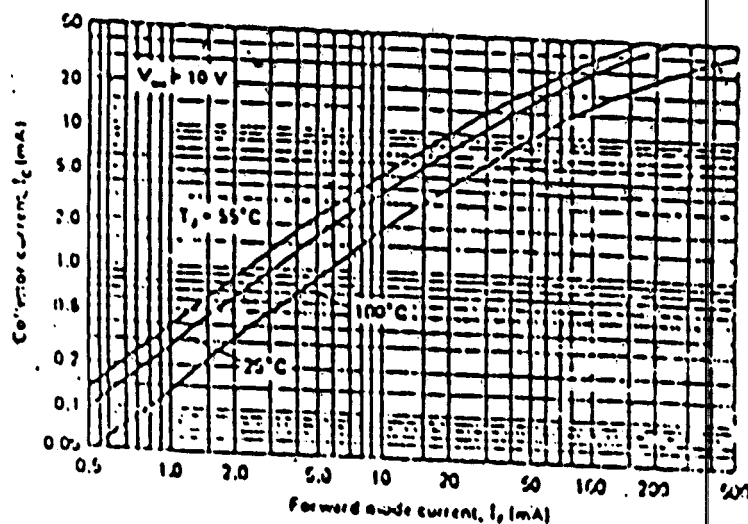
Rangkaian pengubah taraf TTL menjadi keluaran 12 volt diperlihatkan pada gambar 3.7. Cara kerja rangkaian hampir sama dengan rangkaian pengubah taraf 12 volt ke TTL. Dimana rangkaian ini menggunakan optocoupler sebagai

pengisolasi elektris antara sumber dengan beban.

Selain itu ada rangkaian yang menggunakan optocoupler juga tetapi tidak merubah taraf tegangan karena hanya digunakan sebagai sensor untuk mengetahui ada tidaknya benda yang menghalangi cahaya foto diodanya ke fototransistornya. Dimana ini digunakan untuk meng-sensor posisi motor stepper. Yang nantinya berguna untuk penempatan posisi sensor ultrasonik yang tepat yang dipindahkan posisinya dengan bantuan motor stepper.

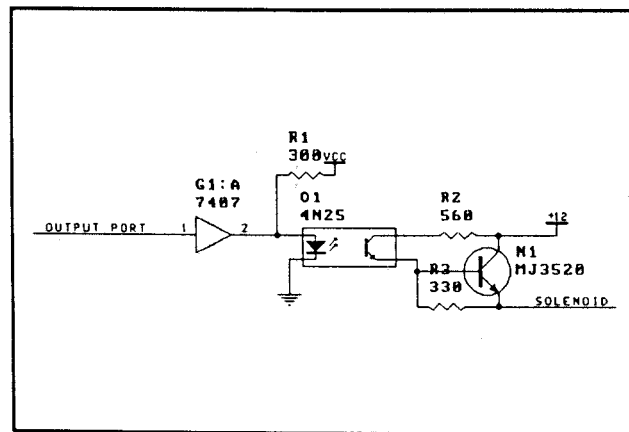


MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER



Gambar 3.6.

KARAKTERISTIK CURRENT TRANSFER RATIO 4N25



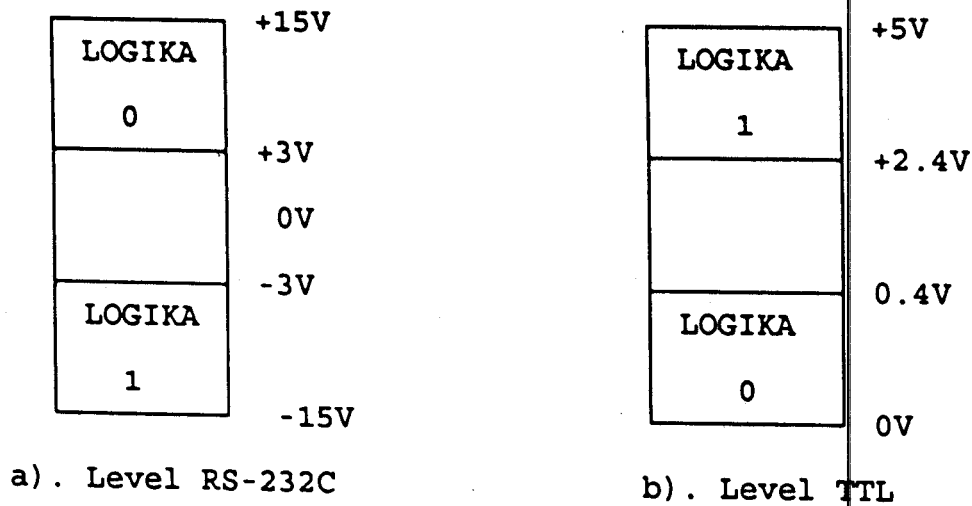
Gambar 3.7.

RANGKAIAN PENGUBAH SINYAL TARAF TTL KE 12V

III.7. RANGKAIAN SERIAL INTERFACE RS-232C

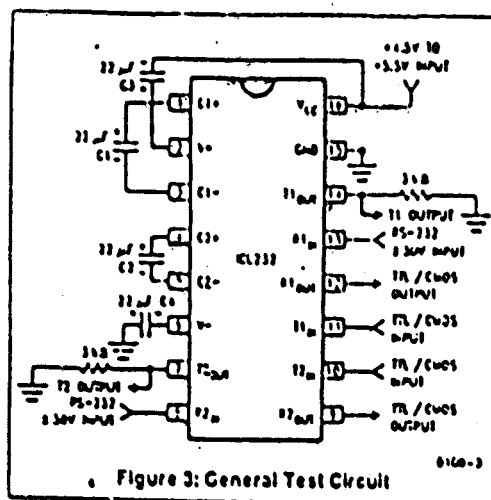
Rangkaian serial interface diperlukan untuk mengubah level TTL ke level RS-232C atau sebaliknya. Perbandingan spesifikasi level sinyal RS-232C dengan TTL dapat dilihat pada gambar 3.8. Kedua standar level sinyal ini berbeda dalam hal pendefinisian level logikanya. Pada level TTL logika '0' didefinisikan pada level 0 - 0,4 volt, logika '1' didefinisikan pada level (-3V) - (-15V). Pada sistem yang direncanakan data serial ditransmisikan melalui kabel transmisi pada level standar RS-232C, sedangkan level yang berhubungan dengan mikrokontroler atau peripheralnya adalah standar level TTL. Oleh karena itu sebelum data serial diterima oleh CPU atau peripheralnya, data tersebut harus diubah terlebih dahulu ke level RS-232C. Untuk mengubah level RS-232C ke level TTL dan sebaliknya digunakan IC ICL232. IC ICL232 ini mempunyai dua buah pengubah level TTL ke level RS-232C dan sebaliknya. Rangkaianannya dapat dilihat pada gambar 3.9.

Port komputer yang digunakan untuk komunikasi dengan peralatan yang direncanakan adalah port serial RS-232C. Port serial ini merupakan DB konektor 25 pin. Hubungan port serial RS-232C komputer dapat dilihat pada Gambar 3.10.



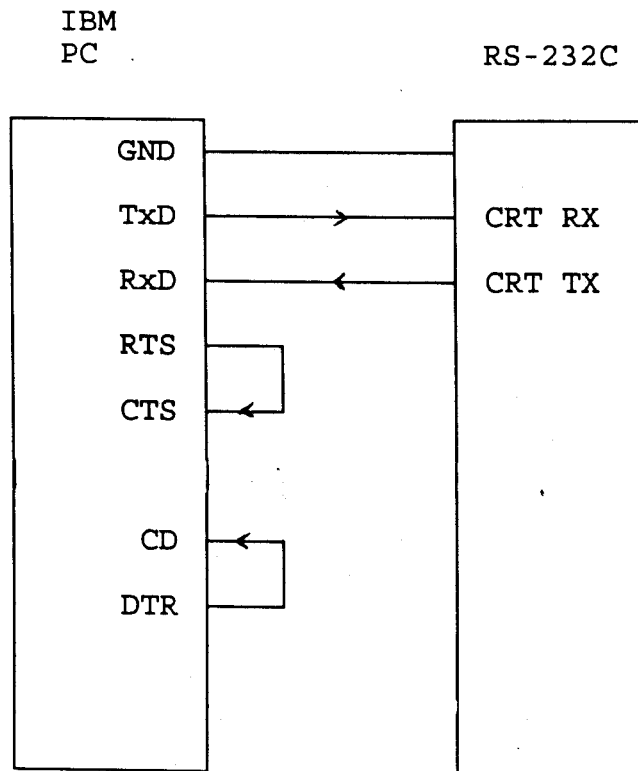
Gambar 3.8.

Perbandingan spesifikasi level RS-232C dengan TTL



Gambar 3.9.

Rangkaian pengubah level



Gambar 3.10.

Hubunga Port serial RS-232C

III.8. RANGKAIAN MOTOR STEPPER

Urutan logika yang diberikan pada motor stepper diperoleh dari port 1 mikrokontroller dengan terlebih dahulu melalui rangkaian driver motor stepper untuk dapat memberikan sumber tegangan 24 volt untuk motor stepper.

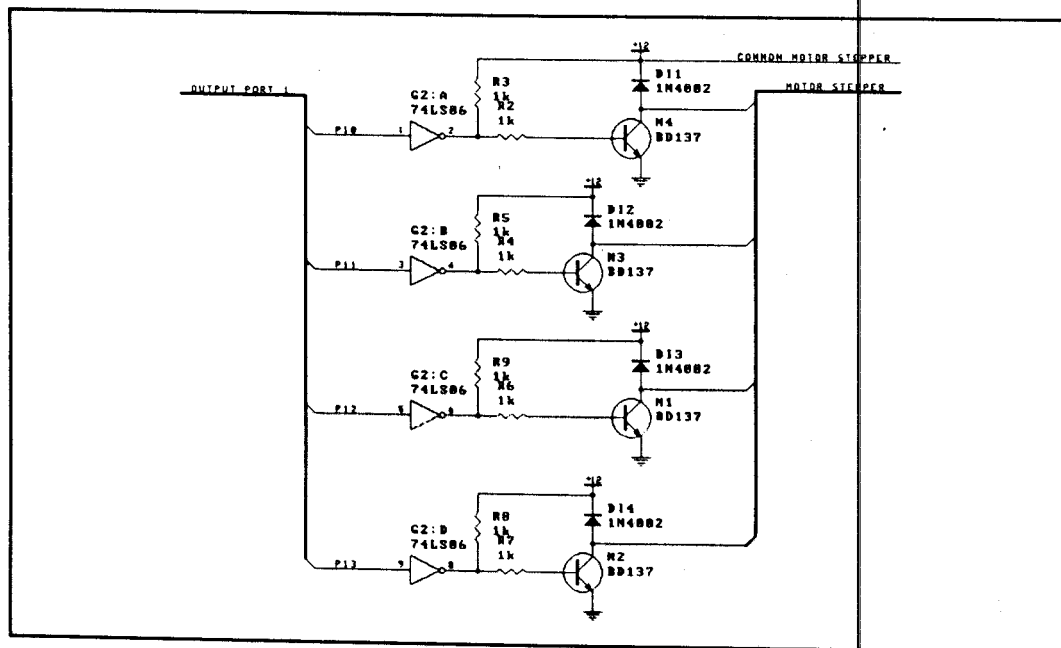
Rangkaian driver untuk motor stepper dapat dilihat pada gambar 3.11.

Adapun tabel kebenaran urutan eksitasi untuk memutar motor stepper secara full step yang diperlihatkan pada tabel 3.2.

Tabel 3.2.

TABEL EKSITASI MOTOR STEPPER

Phase	Output Port 1				Kumparan Motor Stepper			
	P1.3	P1.2	P1.1	P1.0	A	B	C	D
Step 1	1	0	0	1	0	1	1	0
Step 2	0	1	0	1	1	0	1	0
Step 3	0	1	1	0	1	0	0	1
Step 4	1	0	1	0	0	1	0	1
Step 5	1	0	0	1	0	1	1	0



Gambar 3.11.

RANGKAIAN DRIVER MOTOR STEPPER

BAB IV

PERENCANAAN PERANGKAT LUNAK

IV.1. PENDAHULUAN

Pembahasan mengenai perencanaan perangkat lunak ini akan diawali dengan software untuk system 8751h, algoritma dari program untuk mikrokontroller 8751H. Setelah itu dilanjutkan dengan diagram alir dari perangkat lunak pada IBM - PC.

IV.2. Software untuk system 8751h

Mikrokontroller ini dapat menangani operasi aritmetik, baik dalam bentuk binari, BCD maupun dengan pengoperasian bit tunggal yang diperlukan dalam proses kontrol.

IV.2.1. Proses Pemindahan Data

Proses pemindahan data pada mikrokontroller 8751h sangat fleksibel. Sebagai pusat perpindahan data adalah Special Function Register (SFR), yang didalamnya terdapat Akumulator dan register multiguna B yang digunakan untuk proses perpindahan data berupa byte. Selain itu terdapat pula flag C yang terletak di Program Status Word bit 7 yang berfungsi sebagai semacam akumulator bagi pemrosesan data berupa bit. Data dapat ditransfer antara ke-8 register pada Bank register yang sedang digunakan dengan akumulator secara langsung ataupun dengan tujuan di SFR yang dapat dialamati secara langsung. Sedangkan register yang menjadi sumber atau yang menjadi tujuan ditentukan oleh instruksi.

Konstanta yang tersimpan pada memori program dapat dipindahkan secara langsung ke akumulator maupun ke register.

IV.2.2. Pemetaan Memory Program

Lokasi 0 dari memori program digunakan untuk instruksi JMP (jump) yang akan menuju ke program utama. Proses ini dilakukan saat pertama kali power suplai diberikan atau pada tiap kali ada input RESET (switch reset diaktifkan).

Lokasi 023H pada memori program digunakan sebagai vektor alamat untuk interrupt serial. Jadi setiap kali ada interrupt maka alamat ini akan dituju. Lokasi ini berisi AJMP (jump) menuju ke subroutine yang ada. Antara lain subroutine menerima atau mengirim data volume ke Komputer IBM PC.

Program utama dimulai dari alamat 100h yang mana dilanjutkan dengan program-program subroutine.

IV.2.3. Instruksi Pemindahan Data

Instruksi pemindahan data digunakan untuk memindahkan data dari Register, Program Status Word atau alamat byte langsung dari memori program ke akumulator dan sebaliknya. Disamping itu dapat pula digunakan untuk menukar data dari lokasi memori atau register ke Akumulator atau sebaliknya. Instruksi pemindahan data ini dapat juga digunakan untuk memindahkan data immediate ke akumulator, ke memori data, ke alamat byte atau ke register demikian pula sebaliknya. Selain itu instruksi perpindahan data ini juga untuk proses input/output Port, yaitu dengan menindahkan isi Po-P3 di SFR ke akumulator atau

memindahkan data ke P0-P3.

Instruksi-instruksi yang digunakan pada perencanaan software dapat dijelaskan sebagai berikut :

MOV A,Rr : memindahkan data dari register ke
akumulator (dengan $r = 0-7$).

MOV A,@Ri : untuk memindahkan data dari memori data ke
akumulator. Alamat lokasi memori ditunjukkan oleh isi
register (dengan $i = 0-1$).

MOV A,#d : untuk memindahhkan data immediate ke ACC.

MOV A,Direct : untuk memindahkan data di address byte ke
akumulator. (direct menyatakan alamat lokasi data
internal, dapat berupa data RAM atau SFR).

MOV Rr,A : memindahkan data dari akumulator ke
register r.

MOV @Ri,A : memindahkan data dari akumulator ke
memori data dengan alamat ditunjukkan pada register i
(dengan $i = 0$ atau 1).

MOV Rr,#d : memindahkan data immediate ke register r.

MOV @Ri,#d : untuk memindahkan data immediate ke memori
data dengan alamat yang ditunjukkan oleh isi register i.

MOV DPTR,#d : data pointer diisi dengan immediate data.

MOVC A,@A+DPTR : untuk memindahkan code byte yang
ditunjukkan oleh alamat relatif pada DPTR ke akumulator.

IV.2.4. Instruksi Pada Pengoperasian Akumulator

Instruksi yang digunakan pada akumulator adalah instruksi-instruksi yang berhubungan dengan proses logika serta perhitungan matematis yaitu penambahan dengan carry maupun tanpa carry, operasi logika AND, operasi logika OR, dan operasi logika XOR. Disamping itu instruksi ini juga dapat untuk menaikkan atau menurunkan (increment atau decrement) data pada akumulator, mengkomplemen data, menukar nibble, memutar kiri/kanan data, baik dengan carry maupun tidak.

Penjelasan instruksi-instruksi yang digunakan pada akumulator adalah sebagai berikut :

ADD A,#d : untuk menjumlahkan data langsung d, dengan isi akumulator.

ADD A,Direct : untuk menambahkan data di address byte ke akumulator.
(direct menyatakan alamat internal Data RAM atau SFR).

ADD A,Rr : untuk menambahkan isi register r ke akumulator (dengan r = 0-7).

ADD A,@Ri : untuk menambahkan isi lokasi memori yang ditunjukkan oleh Ri ke Akumulator (dengan i = 0 atau 1).

ANL A,Rr : Data di register r di-AND-kan dengan akumulator.

ANL A,#d : Data di akumulator di-AND-kan dengan data immediate d.

ANL A,Direct : Data di Akumulator di-AND-kan dengan data yang ada di address byte.

ORL A,#d : data di akumulator di-OR-kan dengan data immediate d.

ORL A,Direct : data di Akumulator di-OR-kan dengan data yang ada di

address byte.

- INC A : menaikkan 1 data pada akumulator.
- DEC A : menurunkan 1 data di akumulator.
- DA A : untuk mengubah harga bilangan biner pada akumulator menjadi BCD (Binary Coded Desimal).
- SWAP A : untuk menukar nibbles pada akumulator (bit 0-3 diubah menjadi bit 4-7 dan sebaliknya).
- CLR A : isi akumulator di-nolkan.
- CPL A: isi akumulator dikomplemen.
- XCH A,Rr : untuk mengubah isi akumulator dengan isi register r.
- DIV A,B : isi akumulator akan dibagi dengan isi register B, dengan hasilbagi disimpan di akumulator dan sisa pembagi diletakkan diregister B.

IV.2.5. Instruksi untuk Percabangan

Instruksi percabangan dilakukan untuk memanggil subroutine yang ada, melakukan pengetesan kondisi percabangan dan melakukan percabangan sendiri.

Instruksi yang digunakan adalah :

- AJMP Label : Isi program Counter (PC) bit 0-10 diisi dengan lokasi yang ditunjukkan oleh label sehingga pelaksanaan program akan berpindah ke lokasi yang ditunjuk oleh PC, tetapi tidak boleh lebih dari 2 Kbyte dari instruksi AJMP.
- DJNZ Rr,Rel : isi register r akan diturunkan 1 kemudian melakukan percabangan ke lokasi alamat relatif yang ditunjukkan oleh

Rel, selama isi register r tidak sama dengan nol.

DJNZ dir,rel : byte dengan pengalamatan langsung (dir) dikurangi dengan 1, jika tidak sama dengan nol maka dilakukan percabangan ke alamat rel.

JB b,rel : instruksi ini akan melakukan tes terhadap bit b, jika $b=1$ maka PC akan menuju ke alamat relatif rel.

JC rel : proses percabangan terjadi jika carry = 1.

JNC rel : proses percabangan terjadi jika carry = 0.

CJNE Rr,#d,rel : membandingkan isi register r dengan immediate data d jika isi kedua alamat tersebut tidak sama maka PC akan berisi alamat rel dan proses berpindah. Jika $Rr < \#d$ maka carry bit (C) akan diset, sedangkan jika $Rr > \#d$ maka carry = 0. (Dalam hal ini Rr termasuk A dan B).

CJNE A,dir,rel : instruksi ini akan membandingkan isi A dengan alamat byte langsung dir. Jika isi kedua alamat tersebut tidak sama maka PC akan berisi alamat rel dan proses berpindah. Jika $A < dir$ maka carry bit (C) akan diset, sedangkan jika $A > dir$ maka carry = 0.

ACALL add11 : akan mengeksekusi subroutine yang ditunjukkan oleh label tetapi subroutine ini harus dalam jangkauan 2 Kbyte.

RET / RETI : Return berfungsi untuk mengakhiri proses subroutine dan RETI digunakan untuk mengakhiri proses subroutine yang terjadi karena adanya interrupt.

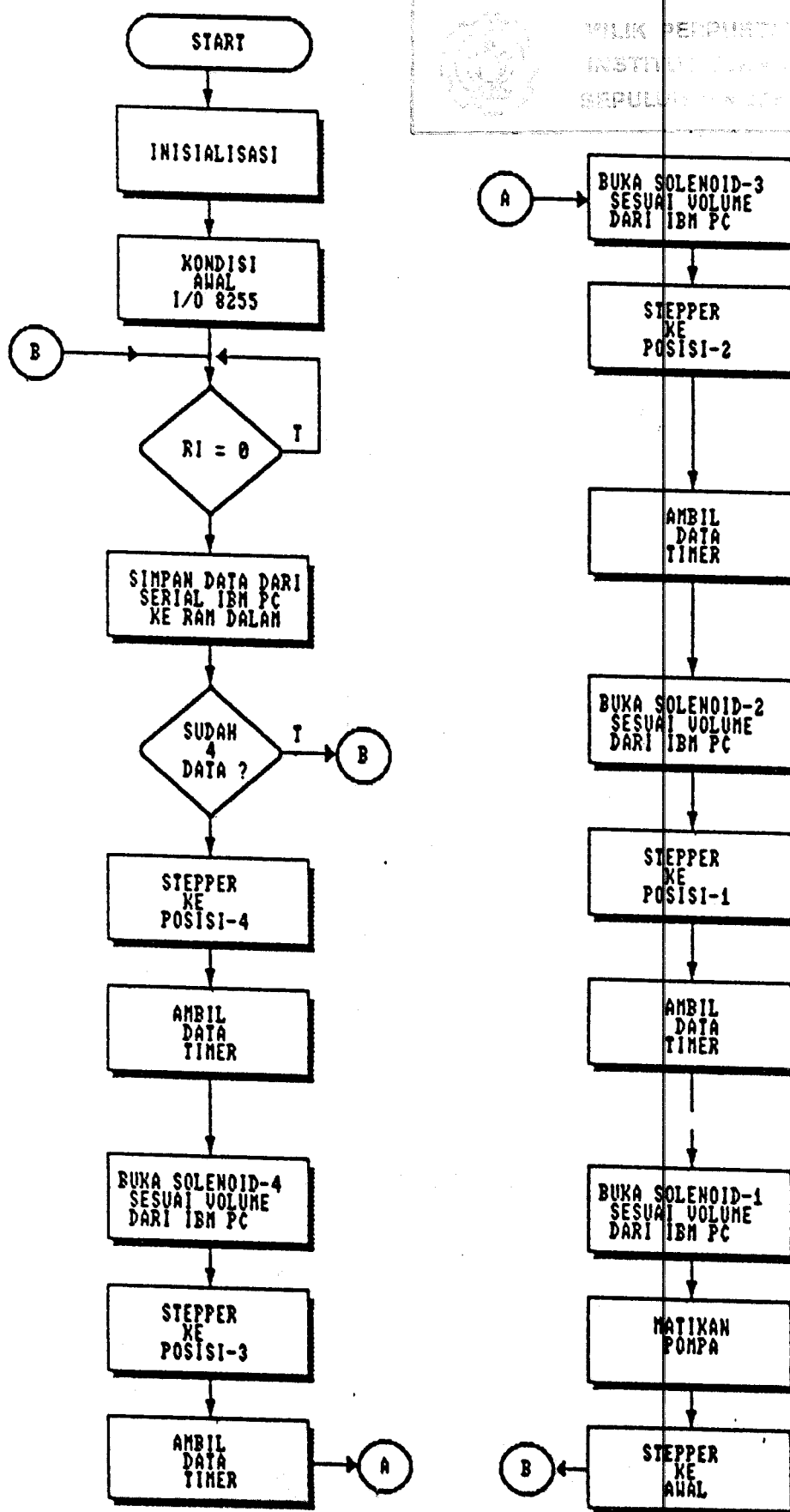
IV.3. PERANGKAT LUNAK MIKROKONTROLLER 8751h.

Flow Chart dari perangkat lunak untuk mikrokontroller 8751H diperlihatkan pada Gambar 4.1. Inisialisasi meliputi mode operasi dari PPI 8255. Pemrosesan masukan/keluaran terdiri dari penyalinan status unit masukan ke RAM dan dari RAM ke unit output yang sesuai.

Status dari semua masukan/keluaran, pewaktu dan data volume dari komputer IBM - PC disimpan pada Memori Data Dalam dari mikrokontroller 8751H, yang dapat dialamati bit secara langsung. Pemetaan dari penggunaan Memori Data Dalam ini diperlihatkan pada Tabel 4.1.

Tabel 4.1. PEMETAAN MEMORI DATA DALAM

	ALAMAT BYTE	
INPUT	20 - 22H	
OUTPUT	23 - 25H	
TIMER	26 - 29H	
DATA VOLUME	2A - 2DH	

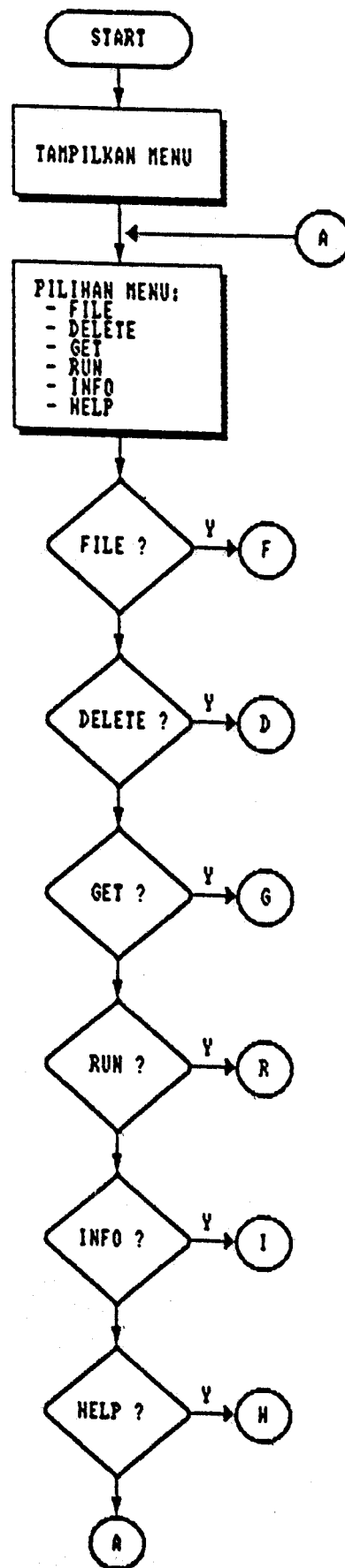


Gambar 4.1. Flow Chart Program Mikrokontroller 8751H

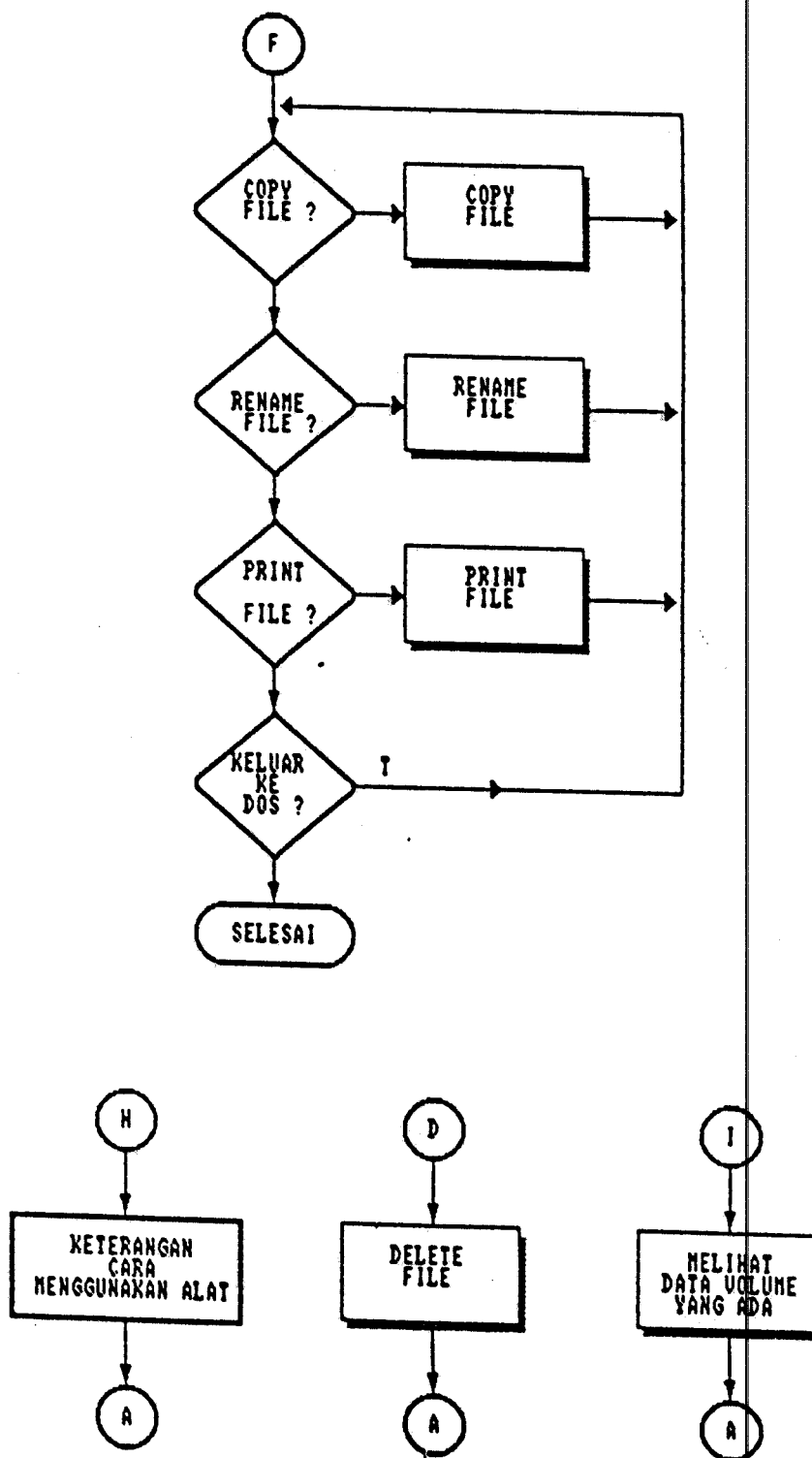
IV.4. PERANGKAT LUNAK PADA IBM - AT

Perangkat lunak pada IBM PC-AT ini menggunakan bahasa pascal. Program tersebut meliputi pembuatan, edit, koreksi, delete komposisi minuman. Selain itu juga dalam hal mengurutkan komposisi minuman yang ada dan terakhir pengiriman data volume minuman secara serial ke mikrokontroller 8751H.

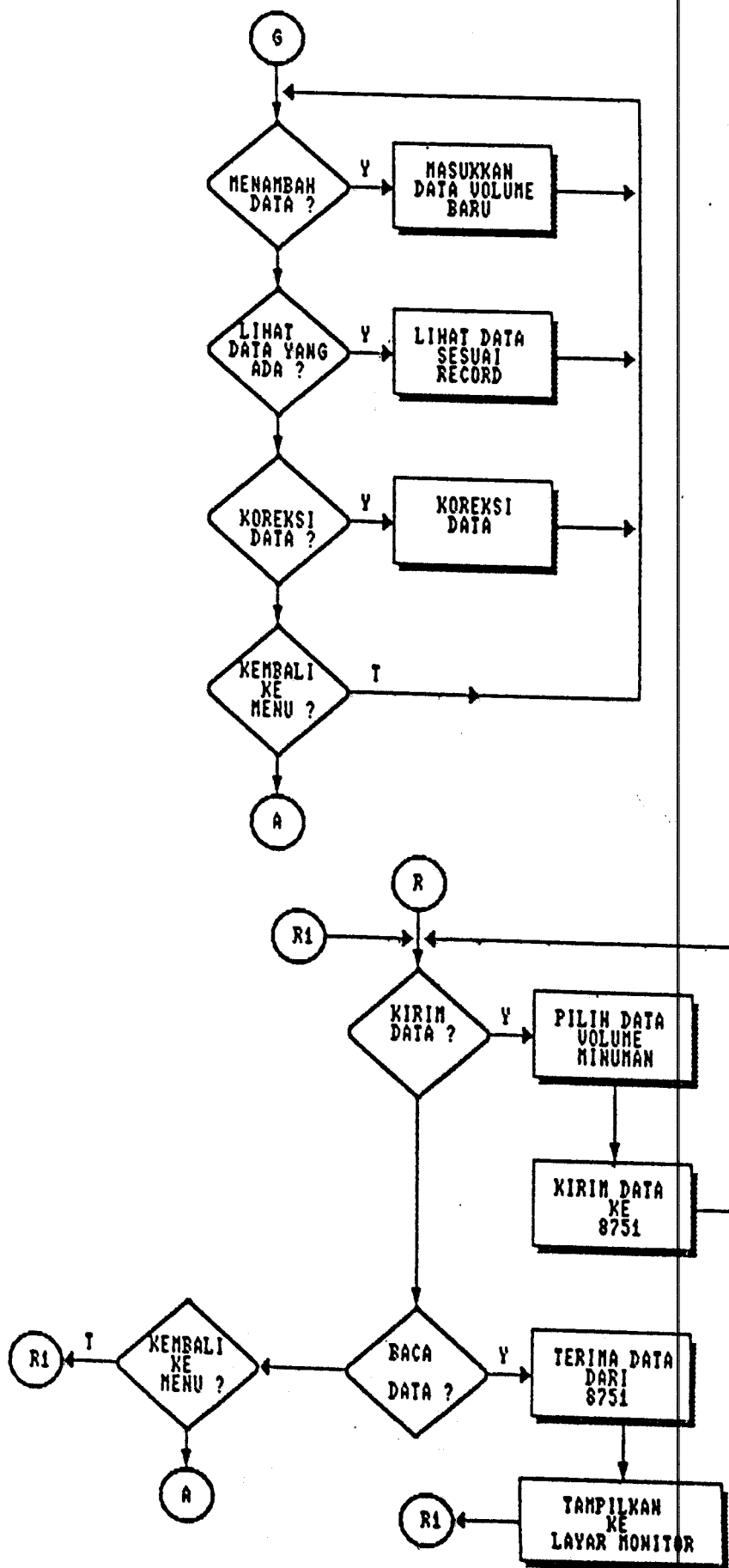
Untuk lebih jelasnya dapat dilihat pada Gambar 4.2. Flow Chart Program Komputer.



Gambar 4.2. Flow Chart Program Komputer



Gambar 4.2. (Lanjutan)
Flow Chart Program Komputer



Gambar 4.2. (Lanjutan)
Flow Chart Program Komputer

BAB V

PENGUJIAN DAN

CARA PENGOPERASIAN ALAT

V.1. PENDAHULUAN

Pengujian peralatan terdiri dari pengujian hubungan komunikasi serial antara mikrokontroler 8751H dengan komputer dan pengujian data volume yang diinginkan yang dipilih lewat komputer dengan yang dikeluarkan oleh alat.

Sedangkan cara pengoperasian alat dijelaskan dengan pembahasan mengenai tiap-tiap menu perangkat lunak terlebih dahulu. Pada bagian berikutnya diambil contoh penerapan untuk menjelaskan cara penggunaannya secara lengkap mulai dari perencanaan pembuatan komposisi minuman baru sampai pengirimannya ke mikrokontroler 8751H untuk diproses menjadi komposisi minuman yang dikehendaki.

V.2. PENGUJIAN KOMUNIKASI SERIAL

Mikrokontroler 8751H mempunyai port serial sehingga tidak dibutuhkan lagi USART untuk merubah data paralel menjadi data serial. Yang diperlukan hanyalah penaik level tegangan yang disesuaikan dengan standar RS-232 yakni dengan menggunakan IC ICL232 (16 pin). Dimana kelebihan ICL232 dibandingkan dengan IC MC1488 dan MC1489 adalah ICL232 mempunyai dua buah receiver (RX), dua buah Transmitter (TX) dan yang lebih penting lagi ICL232 ini cukup diberi tegangan supply +5V.

Sementara itu untuk mengatur baud rate tidak diperlukan eksternal clock karena secara software hal ini dapat dilakukan dengan mudah.

Dari hasil uji coba pengiriman data dari mikrokontroler 8751H ke komputer dengan baud rate sebesar 4800 didapatkan hasil yang memuaskan dalam arti tidak terdapat data yang rusak/hilang selama proses pengiriman data. Sedangkan panjang kabel dilakukan ujicoba sepanjang 2 meter dan mendapatkan hasil yang baik.

V.3. PENGUKURAN OUTPUT ALAT

Pengukuran output alat dilakukan dengan cara membaca hasil volume yang didapat pada gelas, kemudian dibandingkan dengan data volume yang dikehendaki dari komputer IBM PC. Salah satu hasil pengukuran yang didapat dari suatu operasi alat dapat dilihat pada tabel dibawah ini :

Tabel 5.1.

Pengukuran volume dibandingkan dengan data volume dari IBM PC

Data volume dari komputer IBM PC (ml)	Output volume dari alat (ml)	
50	47	
50	48	
50	53	
50	53	

Setelah dilakukan beberapa kali pengujian maka hasil pengukuran yang didapatkan berkisar pada toleransi 5 ml.

V.4. PILIHAN FUNGSI PADA MENU UTAMA

Menu utama dari program yang dibuat diperlihatkan pada Gambar 5.1.

Fungsi dari masing-masing menu tersebut dijelaskan sebagai berikut :

- Help : berisi keterangan dari fungsi-fungsi yang ada sampai cara mengoperasikan alat untuk mendapatkan suatu minuman yang dikehendaki.

- FILE : berisi menu pilihan untuk meng-copy data minuman yang ada dengan nama data minuman baru, mengganti nama file dengan nama file baru, mencetak suatu file data ke printer.

Dan terakhir keluar dari menu utama untuk kembali ke DOS.

- DELETE : menghapus data file komposisi minuman yang ada di disk.

- GET : Menambah, melihat, mengkoreksi data komposisi minuman yang ada di suatu file.

- RUN : Mengirimkan data volume minuman ke mikrokontroller dengan terlebih dahulu memilih data komposisi minuman mana yang akan dikirimkan.

- INFO : Melihat data komposisi minuman yang ada di file. Atau melihat komposisi volume, nama minuman dari suatu data komposisi minuman.

V.5. CONTOH CARA MEMBUAT SUATU KOMPOSISI MINUMAN BARU

Langkah-langkah yang harus dilakukan untuk dapat menghasilkan suatu minuman atau cocktail adalah sebagai berikut :

- Masuk ke program drinking machine, seperti terlihat tampilannya pada gambar 5.1.
- Pilih 'Get'.

Seperti telah dijelaskan pada Bab V.4 yaitu penjelasan dari masing-masing submenu. Dalam hal ini karenan ingin membuat suatu komposisi minuman baru maka dipilih submenu 'Get'.

Pada Submenu 'Get' ini terdiri dari 4 pilihan :

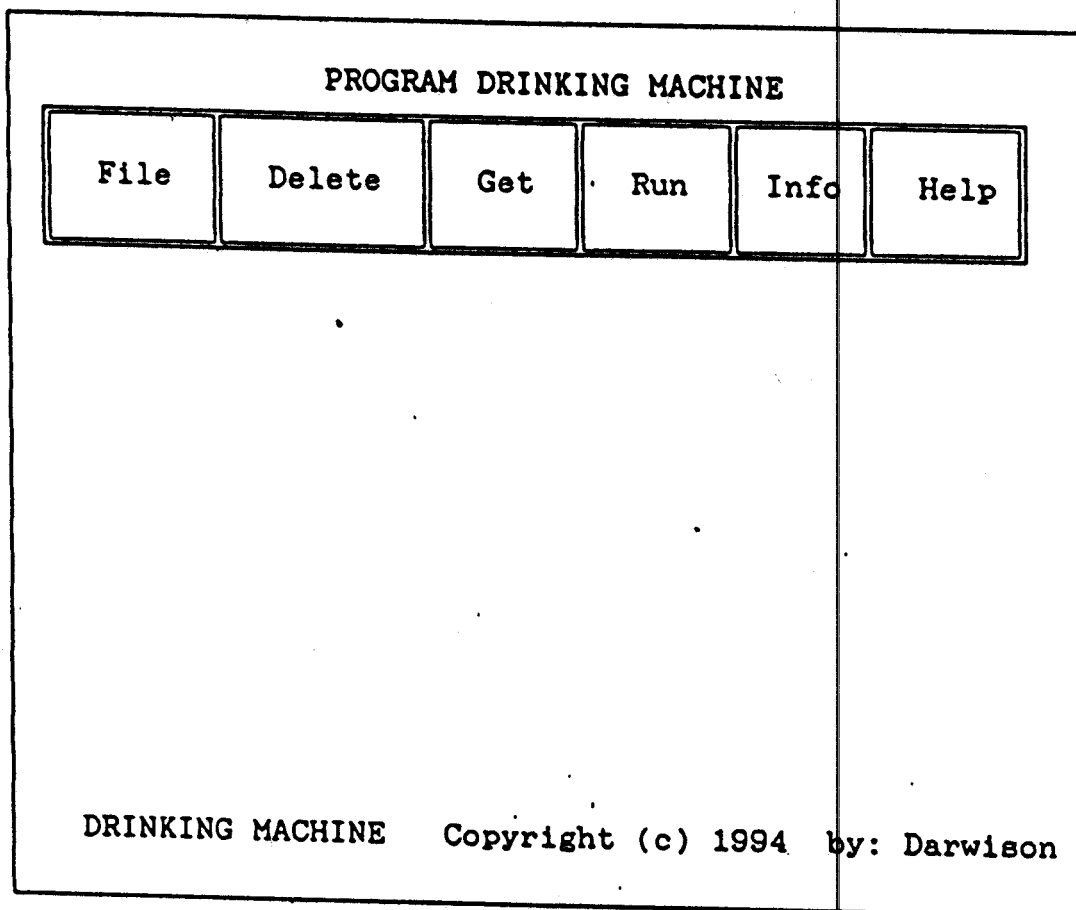
- pilihan 1: yaitu untuk menambah data komposisi minuman yang telah ada di file.
- pilihan 2: yaitu untuk menampilkan data yang ada pada file ke layar monitor.
- pilihan 3: yaitu untuk mengoreksi atau merubah data komposisi minuman yang ada di file.
- pilihan 4: yaitu untuk kembali ke menu utama.

Dalam hal membuat komposisi minuman baru, pilih pilihan 1. Kemudian isi nomor record dari data yang akan dibuat, dengan urutan pengisiannya sebagai berikut:

- Pertama-tama memberi nomor record baru.
- Beri nama cocktailnya.
- Menentukan minuman mana saja yang akan dipakai, beserta jumlah volume yang diinginkan.
- Ulangi sampai nama minuman keempat.
- Jawab pertanyaan:

Jika memilih 'Y' maka akan membuat komposisi minuman baru lagi dan sebaliknya.

Selanjutnya diakhiri dengan memilih pilihan 4 yaitu kembali ke menu utama.



Gambar 5.1.

Tampilan Menu Utama di komputer IBM PC

V.6. CONTOH CARA PENGOPERATIAN

Dengan berpedoman pada penjelasan pada Bab V.4 dan Bab V.5 maka langkah-langkah yang harus dilakukan untuk dapat menghasilkan suatu minuman atau cocktail adalah sebagai berikut :

- Masuk ke program drinking machine.
- Pilih 'Info' untuk melihat komposisi minuman yang akan dikehendaki.
Disini tinggal memilih nomor record untuk mengetahui komposisi minuman suatu cocktail.
- Bila Pilihan nomord record pada Submenu 'Info' telah didapat maka Pilih 'Run' untuk mendapatkan suatu komposisi minuman yang dihendaki atau suatu cocktail, dengan terlebih dahulu memilih submenu 'Kirim data ke Serial' lalu pilih nomor record yang sesuai dari komposisi minuman yang telah dilihat pada submenu 'Info'.
- Selesai, tunggu lagi diproses.
- Kembali ke menu utama.

BAB VI

PENUTUP

VI.1. KESIMPULAN

Dari perencanaan dan pembuatan alat serta pengujian yang telah dilakukan dapat diambil kesimpulan sebagai berikut:

1. Untuk memperbaiki faktor keterbatasan manusia dalam hal mencampur komposisi minuman yang dilakukan langsung oleh manusia, maka dapat dibuatkan suatu alat pencampur otomatis.
2. Untuk mendapatkan volume air tertentu yang mengalir dalam suatu pipa dapat dilakukan dengan bantuan IC mikrokontroller 8751H sebagai kontrol terpusat.
3. Penggunaan mikrokontroller 8751H sebagai unit Pemroses Pusat dibandingkan dengan mikontroller umum seperti Z80 atau 8088, maupun mikrokontroller MCS-48 mempunyai keuntungan-keuntungan antara lain : rangkaian lebih sederhana karena memiliki beberapa komponen di dalamnya (pewaktu/pencacah, RAM, EPROM, port masukan/keluaran paralel maupun serial dan kontroller prioritas interrupt), kecepatan eksekusi yang lebih tinggi dan memiliki instruksi-instruksi untuk melaksanakan operasi logika bit secara langsung.
4. Penggunaan komunikasi data pada mikrokontroller 8751H sangat menguntungkan karena tidak dibutuhkan USART dan pengaturan baud rate cukup secara software dengan memanfaatkan timer yang terdapat dalam

mikrokontroller dan proses pengiriman data pada softwarena relatif sederhana.

VI.2. SARAN-SARAN

Alat yang dibuat ini masih terbuka untuk pengembangan lebih lanjut, terlebih lagi dalam hal menentukan volume suatu cairan tanpa harus mengganggu jalannya cairan itu sendiri.

DAFTAR PUSTAKA

- Ayala, Kenneth J., The 8051 Microcontroller, West Publising Company,
St. Paul, 1991
- Carlin, B., Ultrosonics, McGraw-Hill, 1962
- Coughlin, Robert F. and Driscoll, Frederick F., Operasional Amplifiers and
Linear Integrated Circuits, Pretice Hall of India, India 1985
- Dicky Sarwadi, Bartending, Liberty Yogyakarta, 1987
- Eggebreect, Lewis C., Interfacing to the IBM Personal Computer, Howard W.
Sams & Co., Inc, Indiana Polis, 1983
- HM. Yogianto, Teori dan aplikasi program komputer bahasa Turbo Pascal,
Yogyakarta, 1989
- Warnock, Ian G., Programmable Logic Controller : Operation and
Application, Prentice-Hall International, London, 1988
-, Linear Data Book, National Semiconductor Corporation, Santa Clara,
USA, 1980
-, MCS-51 Family of Single Chip Microcomputer : User's manual, Intel
Corporation, Santa Clara, 1981
-, TTL Data Book, Fairchhild Camera & Instrument Corporation,
California, 1977

```

0000      PA1    EQU    00H      ;INPUT
0001      PB1    EQU    01H      ;INPUT
0002      PC1    EQU    02H      ;INPUT
0003      CW1    EQU    03H
0004      PA2    EQU    04H      ; OUTPUT = solenoid + pompa
0005      PB2    EQU    05H      ; OUTPUT = opto coupler + sensor ultrasonik
0006      PC2    EQU    06H      ;INPUT
0007      CW2    EQU    07H

```

```

0000      org    0000h
0000 020110      jmp    init

```

```

;.....TIMER1 SERVICE ROUTINE.....

```

```

001B      org    001Bh
001B C0D0      push    psw
001D 020100      jmp    vektor

```

```

0100      org    0100h
0100      vektor:
0100 D0D0      pop     psw
0102 32      reti

```

```

;.....PS SERVICE ROUTINE.....

```

```

0023      org    0023h

```

```

0023 C0D0      PUSH    PSW
0025 C0B3      PUSH    DPH
0027 C0B2      PUSH    DPL
0029 C299      CLR     TI
002B C298      CLR     RI
002D D0B2      POP     DPL
002F D0B3      POP     DPH
0031 D0D0      POP     PSW
0033 32      RETI

```

```

0110      org    110h
0110      init:
0110 758150      MOV     SP,#50H      ;inisialisasi stack pointer
0113 75A898      MOV     IE,#10011000B
0116 75BB18      MOV     IP,#00011000B

```

```

;      MOV     DPTR,#0080H      ;alamat awal RAM LUAR
0119 E4      CLR     A
;      MOV     3CH,A
;      MOV     3DH,A
;      PUSH    DPH
;      PUSH    DPL

```

```

011A      SERIAL:
011A 759850      MOV     SCON,#01010000B      ; MODE 1 RECEIVER
011D 758920      MOV     TMOD,#00100000B
0120 438840      ORL     TCON,#40H
0123 758DFA      MOV     TH1,#250      ;SETTING BAUDRATES =4800 bit/s
0126 858D8B      mov     t11,th1
0129 00      NOP
012A 00      NOP

```

```

012B 3098FD      JNB    RI,$
012E E599        MOV    A,SBUF
0130 C298        clr    RI
0132 F52A        mov     2Ah,A      ;...data1

0134 3098FD      JNB    RI,$
0137 E599        MOV    A,SBUF
0139 C298        clr    RI
013B F52B        mov     2Bh,A      ;....data2

013D 3098FD      JNB    RI,$
0140 E599        MOV    A,SBUF
0142 C298        clr    RI
0144 F52C        mov     2Ch,A      ;....data3

0146 3098FD      JNB    RI,$
0149 E599        MOV    A,SBUF
014B C298        clr    RI
014D F52D        mov     2Dh,A      ;.....data4

;.....inisialisasi 8255 & kondisi awalnya .....
014F          AWAL:
014F 1203C8      CALL    DELAY
0152 7803        mov     RO,#CW1
0154 749B        mov     A,#9BH      ;A,B,C INPUT
0156 F2          movx    @RO,A
0157 7807        mov     RO,#CW2
0159 7480        mov     A,#80H      ;A,B OUTPUT DAN C INPUT
015B F2          movx    @RO,A
015C 7805        mov     RO,#PB2
015E 74F1        mov     A,#0F1H     ;CLR=1 , NAND=0 ,OPTOCOUPLER=1 (on)
0160 F2          movx    @RO,A
0161 7804        mov     RO,#PA2
0163 74FF        mov     A,#0FFH     ;solenoid(741s06), pompa(741s07) = off
0165 F2          movx    @RO,A

0166 1203C8      CALL    DELAY
0169 7804        mov     RO,#PA2
016B 743F        mov     A,#3FH      ;hidupkan pompa
016D F2          movx    @RO,A
016E 7590FF      mov     p1,#0ffh    ; matikan stepper

;.....GERAKKAN STEPPER MUNDUR.....
0171 12032C      call    opto4

0174 1203C8      CALL    DELAY
0177 1202D5      call    opto3

017A 1203C8      call    DELAY
017D 12027E      call    opto2

0180 1203C8      call    DELAY
0183 120227      call    opto1

0186 1203BB      call    tunda

```

```

0189 7804      MOV R0,#PA2
018B 74FF      mov A,#0FFH      ;....sol4 = off...., pompa(74ls07) = off
018D F2        movx @R0,A      ;solenoid sensor = off

018E 1203C8     call delay
0191 1201E2     call mundur

0194 1203BB     CALL tunda

0197 E52A      mov A,2Ah      ;...data1
0199 F599      MOV SBUF,A
019B 3099FD     JNB TI,$
019E C299      clr TI
01A0 1203D1     call seri

01A3 E52B      mov A,2Bh      ;...data2
01A5 F599      MOV SBUF,A
01A7 3099FD     JNB TI,$
01AA C299      clr TI
01AC 1203D1     call seri

01AF E52C      mov A,2Ch      ;...data3
01B1 F599      MOV SBUF,A
01B3 3099FD     JNB TI,$
01B6 C299      clr TI
01B8 1203D1     call seri

01BB E52D      mov A,2Dh      ;...data4
01BD F599      MOV SBUF,A
01BF 3099FD     JNB TI,$
01C2 C299      clr TI
01C4 1203D1     call seri

01C7 1203BB     call tunda
01CA 1203BB     call tunda
01CD 1203BB     CALL tunda
01D0 1203BB     call tunda
01D3 1203BB     call tunda
01D6 1203BB     call tunda
01D9 1203BB     call tunda
01DC 1203BB     call tunda

01DF 02014F     ljmp ANAL

01E2          mundur:
01E2 7A4F      mov r2,#4fh
01E4          mun:
01E4 1203C8     call delay
01E7 7590AA     mov pl,#0aah
01EA 1203C8     call delay
01ED 759099     mov pl,#99h
01F0 1203C8     call delay
01F3 759055     mov pl,#55h

```

```

01F6 1203C8      call delay
01F9 759066      mov pl,#66h

01FC 7801      MOV R0,#PB1      ;BACA DATA POSISI STEPPER
01FE E2      MOVX A,@R0
01FF 30E702      jnb acc.7,tambah

0202 DAE0      djnz r2,mun
0204      tambah:
0204 7904      mov r1,#04h
0206      tam:
0206 1203C8      call delay
0209 7590AA      mov pl,#0aah
020C 1203C8      call delay
020F 759099      mov pl,#99h
0212 1203C8      call delay
0215 759055      mov pl,#55h
0218 1203C8      call delay
021B 759066      mov pl,#66h
021E D9E6      djnz r1,tam

0220 7590FF      mov pl,#0ffh ; matikan stepper

0223 1203C8      call delay
0226 22      ret

0227      optol:
0227 7A4F      mov r2,#4fh
0229      ulang1:
0229 1203C8      call delay
022C 759066      mov pl,#66h
022F 1203C8      call delay
0232 759055      mov pl,#55h
0235 1203C8      call delay
0238 759099      mov pl,#99h
023B 1203C8      call delay
023E 7590AA      mov pl,#0aah

0241 7801      MOV R0,#PB1      ;BACA DATA POSISI STEPPER
0243 E2      MOVX A,@R0
0244 30E402      jnb acc.4,henti4

0247 DAE0      djnz r2,ulang1
0249      henti4:
0249 7590FF      mov pl,#0ffh ; matikan stepper

024C 1203C8      call delay

024F 7804      mov R0,#PA2
0251 742F      mov A,#2FH      ;solenoid(741s06) = off, pompa(741s07) = on
0253 F2      movx @R0,A      ;.....solenoid sensor = on.....

0254 7805      mov R0,#PB2
0256 74F0      mov A,#0F0H      ;...reset 7474 & 393..., NAND=0 ,OPTOCOUPLER=1 (on)
0258 F2      movx @R0,A

```



```

0259 7805      mov R0,#PB2
025B 74F1      mov A,#0F1H      ;CLR=1 , NAND=0 ,OPTOCOUPLER=1 (on)
025D F2       movx @R0,A
025E 1203BB    call tunda
0261 7805      mov R0,#PB2
0263 74F3      mov A,#0F3H      ;CLR=1 ,... counter aktif... ,OPTOCOUPLER=1 (on)
0265 F2       movx @R0,A

0266 7805      mov R0,#PB2
0268 74F1      mov A,#0F1H      ;CLR=1 ,... counter berhenti... ,OPTOCOUPLER=1 (on)
026A F2       movx @R0,A

026B 7804      mov R0,#PA2
026D 743F      mov A,#3FH       ;solenoid(74ls06) = off, pompa(74ls07) = on
026F F2       movx @R0,A       ;.....solenoid sensor = off.....

; .... LAMA AKTIF SOLENOID....DGN LOOPING R2...
0270 7804      mov R0,#PA2
0272 743B      mov A,#3BH       ;....soli = ON...., pompa(74ls07) = on
0274 F2       movx @R0,A       ;solenoid sensor = off

0275 120383    call air1

0278 7804      mov R0,#PA2
027A 743F      mov A,#3FH       ;....soli = off...., pompa(74ls07) = off
027C F2       movx @R0,A       ;solenoid sensor = off

027D 22        ret

027E          opto2:
027E 7A4F      mov r2,#4fh
0280          ulang2:
0280 1203C8    call delay
0283 759066    mov p1,#66h
0286 1203C8    call delay
0289 759055    mov p1,#55h
028C 1203C8    call delay
028F 759099    mov p1,#99h
0292 1203C8    call delay
0295 7590AA    mov p1,#0aah

0298 7801      mov R0,#PB1      ;BACA DATA POSISI STEPPER
029A E2       movx A,@R0
029B 30E502    jnb acc.5,henti5

029E DAE0      djnz r2,ulang2
02A0          henti5:
02A0 7590FF    mov p1,#0ffh      ; matikan stepper

02A3 1203C8    call delay

02A6 7804      mov R0,#PA2
02AB 742F      mov A,#2FH       ;solenoid(74ls06) = off, pompa(74ls07) = on
02AA F2       movx @R0,A       ;.....solenoid sensor = on.....

```

```

02AB 7805      mov RO,#PB2
02AD 74F0      mov A,#0F0H      ;...reset 7474 & 393..., NAND=0 ,OPTOCOUPLER=1 (on)
02AF F2        movx @R0,A
02B0 7805      mov RO,#PB2
02B2 74F1      mov A,#0F1H      ;CLR=1 , NAND=0 ,OPTOCOUPLER=1 (on)
02B4 F2        movx @R0,A
02B5 1203BB    call tunda
02B8 7805      mov RO,#PB2
02BA 74F3      mov A,#0F3H      ;CLR=1 ,... counter aktif... ,OPTOCOUPLER=1 (on)
02BC F2        movx @R0,A

02BD 7805      mov RO,#PB2
02BF 74F1      mov A,#0F1H      ;CLR=1 ,... counter berhenti... ,OPTOCOUPLER=1 (on)
02C1 F2        movx @R0,A

02C2 7804      mov RO,#PA2
02C4 743F      mov A,#3FH      ;solenoid(74ls06) = off, pompa(74ls07) = on
02C6 F2        movx @R0,A      ;.....solenoid sensor = off.....

; .... LAMA AKTIF SOLENOID....DGN LOOPING R2...
02C7 7804      mov RO,#PA2
02C9 7437      mov A,#37H      ;....sol2 = ON...., pompa(74ls07) = on
02CB F2        movx @R0,A      ;solenoid sensor = off

02CC 120391    call air2

02CF 7804      mov RO,#PA2
02D1 743F      mov A,#3FH      ;....sol2 = off...., pompa(74ls07) = off
02D3 F2        movx @R0,A      ;solenoid sensor = off

02D4 22        ret

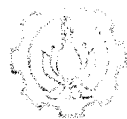
02D5          opto3:
02D5 7A4F      mov r2,#4fh
02D7          ulang3:
02D7 1203C8    call delay
02DA 759066    mov p1,#66h
02DD 1203C8    call delay
02E0 759055    mov p1,#55h
02E3 1203C8    call delay
02E6 759099    mov p1,#99h
02E9 1203C8    call delay
02EC 7590AA    mov p1,#0aah

02EF 7801      mov RO,#PB1      ;BACA DATA POSISI STEPPER
02F1 E2        movx A,@R0
02F2 30E602    jnb acc.6,henti6

02F5 DAE0      djnz r2,ulang3
02F7          henti6:
02F7 7590FF    mov p1,#0ffh      ; matikan stepper

02FA 1203C8    call delay

```



MILIK PERPUSTAKAAN
 INSTITUT TEKNOLOGI
 SEPULUH - NOPEMBER

```

02FD 7804      mov R0,#PA2
02FF 742F      mov A,#2FH      ;solenoid(74ls06) = off, pompa(74ls07) = on
0301 F2        movx @R0,A      ;.....solenoid sensor = on.....

0302 7805      mov R0,#PB2
0304 74F0      mov A,#0F0H      ;...reset 7474 & 393..., NAND=0 ,OPTOCOUPLER=1 (on)
0306 F2        movx @R0,A
0307 7805      mov R0,#PB2
0309 74F1      mov A,#0F1H      ;CLR=1 , NAND=0 ,OPTOCOUPLER=1 (on)
030B F2        movx @R0,A
030C 1203BB    call tunda
030F 7805      mov R0,#PB2
0311 74F3      mov A,#0F3H      ;CLR=1 ,... counter aktif... ,OPTOCOUPLER=1 (on)
0313 F2        movx @R0,A

0314 7805      mov R0,#PB2
0316 74F1      mov A,#0F1H      ;CLR=1 ,... counter berhenti... ,OPTOCOUPLER=1 (on)
0318 F2        movx @R0,A

0319 7804      mov R0,#PA2
031B 743F      mov A,#3FH      ;solenoid(74ls06) = off, pompa(74ls07) = on
031D F2        movx @R0,A      ;.....solenoid sensor = off.....

; .... LAMA AKTIF SOLENOID....DGN LOOPING R2...
031E 7804      mov R0,#PA2
0320 743D      mov A,#3dh      ;....sol3 = ON....., pompa(74ls07) = on
0322 F2        movx @R0,A      ;solenoid sensor = off

0323 12039F    call air3

0326 7804      mov R0,#PA2
0328 743F      mov A,#3FH      ;....sol3 = off....., pompa(74ls07) = off
032A F2        movx @R0,A      ;solenoid sensor = off

032B 22        ret

032C          opto4:
032C 7A4F      mov r2,#4fh
032E          ulang4:
032E 1203C8    call delay
0331 759066    mov p1,#66h
0334 1203C8    call delay
0337 759055    mov p1,#55h
033A 1203C8    call delay
033D 759099    mov p1,#99h
0340 1203C8    call delay
0343 7590AA    mov p1,#0aah

0346 7801      mov R0,#PB1      ;BACA DATA POSISI STEPPER
0348 E2        movx A,@R0
0349 30E702    jnb acc.7,henti7

034C DAE0      djnz r2,ulang4
034E          henti7:

```

```

034E 7590FF      mov  p1,#0ffh  ; matikan stepper
0351 1203C8      call delay
0354 7804        mov  R0,#PA2
0356 742F        mov  A,#2FH      ;solenoid(741s06) = off, pompa(741s07) = on
0358 F2          movx @R0,A      ;.....solenoid sensor = on.....

0359 7805        mov  R0,#PB2
035B 74F0        mov  A,#0F0H      ;...reset 7474 & 393..., NAND=0 ,OPTOCOUPLER=1 (on)
035D F2          movx @R0,A
035E 7805        mov  R0,#PB2
0360 74F1        mov  A,#0F1H      ;CLR=1 , NAND=0 ,OPTOCOUPLER=1 (on)
0362 F2          movx @R0,A
0363 1203BB      call tunda
0366 7805        mov  R0,#PB2
0368 74F3        mov  A,#0F3H      ;CLR=1 ,... counter aktif... ,OPTOCOUPLER=1 (on)
036A F2          movx @R0,A

036B 7805        mov  R0,#PB2
036D 74F1        mov  A,#0F1H      ;CLR=1 ,... counter berhenti... ,OPTOCOUPLER=1 (on)
036F F2          movx @R0,A

0370 7804        mov  R0,#PA2
0372 743F        mov  A,#3FH      ;solenoid(741s06) = off, pompa(741s07) = on
0374 F2          movx @R0,A      ;.....solenoid sensor = off.....

; .... LAMA AKTIF SOLENOID....DGN LOOPING R2...
0375 7804        mov  R0,#PA2
0377 743E        mov  A,#3eH      ;....sol4 = DN....., pompa(741s07) = on
0379 F2          movx @R0,A      ;solenoid sensor = off

037A 1203AD      call air4

037D 7804        mov  R0,#PA2
037F 743F        mov  A,#3FH      ;....sol4 = off....., pompa(741s07) = off
0381 F2          movx @R0,A      ;solenoid sensor = off

0382 22          ret

0383            air1:
0383 E52A        mov  a,2ah ;.....data1
0385 FB          mov  r3,A
0386            a2:
0386 7C0E        mov  r4,#0eh

0388            a3:
0388 7FF0        mov  r7,#0f0h
038A DFFE        djnz r7,$
038C DCFA        djnz r4,a3
038E DBF6        djnz r3,a2
0390 22          ret

0391            air2:

```

```

0391 E52B      mov a,2bh ; ....data2
0393 FB        mov r3,A
0394          a22:
0394 7C0E      mov r4,#0eh

```

```

0396          a32:
0396 7FF0      mov r7,#0f0h
0398 DFFE      djnz r7,$
039A DCFA      djnz r4,a32
039C DBF6      djnz r3,a22
039E 22        ret

```

```

039F          air3:
039F E52C      mov a,2ch ;...data3
03A1 FB        mov r3,A
03A2          a23:
03A2 7C0E      mov r4,#0eh

```

```

03A4          a33:
03A4 7FF0      mov r7,#0f0h
03A6 DFFE      djnz r7,$
03A8 DCFA      djnz r4,a33
03AA DBF6      djnz r3,a23
03AC 22        ret

```

```

03AD          air4:
03AD E52D      mov a,2dh ;...data4
03AF FB        mov r3,A
03B0          a24:
03B0 7C0E      mov r4,#0eh

```

```

03B2          a34:
03B2 7FF0      mov r7,#0f0h
03B4 DFFE      djnz r7,$
03B6 DCFA      djnz r4,a34
03B8 DBF6      djnz r3,a24
03BA 22        ret

```

```

03BB          tunda:
03BB 7B10      mov r3,#10h
03BD          tunda1:
03BD 7CFF      mov r4,#0ffh

```

```

03BF          tunda2:
03BF 7FFF      mov r7,#0ffh
03C1 DFFE      djnz r7,$
03C3 DCFA      djnz r4,tunda2
03C5 DBF6      djnz r3,tunda1
03C7 22        ret

```

```

03C8          delay:
03C8 7C05      mov r4,#05h
03CA          loop1:
03CA 7FFF      mov r7,#0ffh
03CC DFFE      djnz r7,$

```

```
03CE DCFA      djnz r4,loop1
03D0 22        ret

03D1          seri:
03D1 7CFF      mov r4,#0ffh
03D3          seril:
03D3 7FFF      mov r7,#0ffh
03D5 DFFE      djnz r7,$
03D7 DCFA      djnz r4,seril
03D9 22        ret

0110          END INIT
```

***** SYMBOL REFERENCE TABLE *****

03B6 = A2	0394 = A22	03A2 = A23
03B0 = A24	0388 = A3	0396 = A32
03A4 = A33	03B2 = A34	00E0 = ACC
03B3 = AIR1	0391 = AIR2	039F = AIR3
03AD = AIR4	014F = AWAL	0003 = CW1
0007 = CW2	03C8 = DELAY	0083 = DPH
0082 = DPL	0249 = HENTI4	02A0 = HENTI5
02F7 = HENTI6	034E = HENTI7	00A8 = IE
0110 = INIT	00B8 = IP	03CA = LOOP1
01E4 = MUN	01E2 = MUNDUR	0227 = OPT01
027E = OPT02	02D5 = OPT03	032C = OPT04
0090 = P1	0000 = PA1	0004 = PA2
0001 = PB1	0005 = PB2	0002 = PC1
0006 = PC2	00D0 = PSM	0098 = RI
0099 = SBUF	0098 = SCON	03D1 = SERI
03D3 = SERI1	011A = SERIAL	0081 = SP
0206 = TAM	0204 = TAMBAH	0088 = TCON
008D = TH1	0099 = TI	008B = TL1
0089 = TMO0	03BB = TUNDA	03BD = TUNDA1
03BF = TUNDA2	0229 = ULANG1	0280 = ULANG2
02D7 = ULANG3	032E = ULANG4	0100 = VEKTOR

TOTAL SYMBOLS DEFINED = 63

NO ERROR(S) FOUND

END OF ASSEMBLY

07-25-1994

21:40:31

Uses Crt,Dos,Graph,Printer,ShortCt,TVisi;

label looping;

var mnu : string;
Chrs : char;
k,scan: integer;
Reg : Registers;
Baris, Kolom : Integer;

(\$I Type7.inc)
(\$I Lihrec.inc)
(\$I Untype2.inc)
(\$I Untype3.inc)
(\$I Untype4.inc)
(\$I teks6.inc)
(\$I mouse9.inc)
(\$I tx14.inc)
(\$I rx14.inc)

Procedure DeclareParent;
begin

{mouse9;}

mnu:='&#File &#Delete &#Get &#Run &#Info &#Help&';
visimenuesign('Program Drinking Machine ',mnu,(16*4)+15);

box(1,23,80,25,(16*7)+1,True);
window(2,24,79,25);textattr:=16*7+1;
Write(#0,' DRINKING MACHINE Copyright (c) 1994 by : ');
textcolor(16);
write(#0#0,' AMI ');
window(1,1,80,25);
end;

Procedure ScanKey;
begin

Repeat;
repeat;
chrs:=readkey;
until chrs in[AltF,AltD,AltG,AltR,AltI,AltM,AltH];

if Chrs=AltF Then begin
mnu:='&Copy&&Rename&&Print&&Exit&';
submenus(menuplace[1],mnu,scan);
if scan=1 Then begin
msgbox(' Copy ',1,scan);
if scan=0 then begin

untype4;

end
else
window(1,1,80,25);

begin
declareparent;
scankey;
end;
end;


```

if scan=2 then begin
msgbox('      Rename      ',1,scan);
if scan=0 then begin

untype3;

end
else
window(1,1,80,25);

begin
declareparent;
scankey;
end;
end;

if scan=3 Then begin
msgbox('      Print      ',1,scan);
if scan=0 then begin

teks6;

end
else
window(1,1,80,25);

begin
declareparent;
scankey;
end;
end;

if scan=4 then begin
msgbox('Do you want to end this program ?',1,scan);
if scan=0 then begin
textattr:=7;
clrscr;
writeln('Say Thanks to Awi!');
halt;
end;
end;
end;

if Chrs=AltD Then begin
mnu:='&Delete file data ?&';
submenus(menuplace[2],mnu,scan);
if scan=1 then begin
mnu:=' Delete file data      ';msgbox(mnu,1,scan);
if scan=0 then begin

untype2;

end
else
window(1,1,80,25);

begin
declareparent;
scankey;
end;
end;

```

```

end;

if Chrs=AltG Then begin
  mnu:='&Buat kreasi minuman baru&';
  submenus(menuplace[3],mnu,scan);
  if scan=1 then begin
    msgbox('    New Cocktail    ',0,scan);

    type7;

    window(1,1,80,25);
    end;

    begin
    declareparent;
    scankey;
    end;
    end;

    if Chrs=AltR then begin
      mnu:='&Kirimdata ke Port Serial&&Baca data dari Port Serial&';
      submenus(menuplace[4],mnu,scan);
      if scan=1 then begin
        msgbox('        Ready        ',1,scan);
      if scan=0 then begin

        tx14;

        end
        else
        window(1,1,80,25);
        begin
        declareparent;
        scankey;
        end;
        end;

        if scan=2 then begin
          msgbox('        Ready        ',1,scan);
          if scan=0 then begin

            rx14;

            end
            else
            window(1,1,80,25);
            begin
            declareparent;
            scankey;
            end;
            end;

            end;

            if Chrs=AltI then begin
              mnu:='&Nama Cocktail yang ada&';
              submenus(menuplace[5],mnu,scan);
              if scan=1 then begin
                mnu:=' Nama cocktail ';msgbox(mnu,0,scan);

                lihrec;

                end

```

```
else  
window(1,1,80,25);  
begin  
declareparent;  
scankey;  
end;  
end;
```

```
if Chrs=AltH Then begin  
mnu:='&Tentang Program &&Tentang Programmer&';  
submenus(menuplace[6],mnu,scan);  
if scan=1 then begin  
mnu:='Drinking Machine, adalah program untuk menjalankan alat lewat komputer';  
msgbox(mnu,0,scan);  
end else begin  
if scan =2then begin  
mnu:='Drinking Machine ini ciptakan oleh Awi';  
msgbox(mnu,0,scan);  
end;  
end;  
end;  
Until Chrs=AltP;  
end;  
begin  
declareparent;  
scankey;  
end.
```

```
Procedure type7;  
{Program Integrasi_dengan_menu_pilihan;  
Uses Crt;}
```

```
Procedure MenambahData;  
{I type27.inc}
```

```
Procedure CariBerdasarkanrecord;  
{I Type417.inc}
```

```
Procedure KoreksiBerdasarkanRecord;  
{I Type517.inc}
```

```
(* PROGRAM UTAMA MENAMPILKAN MENU PILIHAN *)
```

```
Var  
  Pilih :Byte;  
Begin  
  While True do  
    begin  
      clrscr;  
      GotoXY(15, 5); Write('      <<<<      MENU      >>>>');  
      GotoXY(15, 7); Write('1. Menambah data');  
      GotoXY(15, 8); Write('2. Menampilkan data tertentu berdasarkan record');  
      GotoXY(15, 9); Write('3. Koreksi data tertentu berdasarkan record');  
      GotoXY(15,10); Write('4. Selesai ...');  
  
      Pilih := 0;  
      While (Pilih < 1) or (Pilih > 7) do  
        begin  
          GotoXY(15,20);Write('Pilih nomor (1-4) ? ');Readln(Pilih);  
          If (Pilih <1) or (Pilih > 4) then  
            Write('^G');  
        end;  
  
      Case Pilih of  
        1: MenambahData;  
        2: CariBerdasarkanRecord;  
        3: KoreksiBerdasarkanRecord;  
        4: Exit;  
      end;  
    end;  
  end;  
end; {menu}
```

```
{Procedure tx14;}
Uses
    Dos,Crt,Printer;
Const
    Head: Array[1..6] of string = ('      Kirim Data ke Port Serial
                                     '
                                     '
                                     '
                                     '
                                     '
                                     '      Kembali ke menu Utama
                                     ');
Type
    pixel = Record
        Karakter : char;
        Atribut  : byte;
    end;
    BufferLayar = Array[1..25,1..80] of pixel;
    BufferSeri  = Array[1..12] of byte;
    BufferSeril = Array[1..512] of real;
Var
    ch,K       : Char;
    index      : word;
    Baris,by,Kolom : Byte;
    Reg        : Registers;
    Screen     : BufferLayar Absolute $B800:$0000;
    ScreenSave : Array[1..25,1..80] of pixel;
    buffers    : BufferSeri;
    XReal      : BufferSeril;

Procedure Pilih_Tulisan(Pilih: integer);
begin
    Case Pilih of
        8 : Begin GotoXY(22,8) ; Write(Head[1]); end;
        9 : Begin GotoXY(22,9) ; Write(Head[2]); end;
        10 : Begin GotoXY(22,10); Write(Head[3]); end;
        11 : Begin GotoXY(22,11); Write(Head[4]); end;
        12 : Begin GotoXY(22,12); Write(Head[5]); end;
        13 : Begin GotoXY(22,13); Write(Head[6]); end;
    end;
end;

Procedure Tampak;
begin
    Reg.AH := $01;
    Reg.CH := $06;
    Reg.CL := $07;
    Intr($10,Reg);
end;

Procedure Pesan;
begin
    gotoXY(23,22);Write('      File tidak ditemukan   ');
    delay(1500);
end;

Function Transmittan(by:integer):real;
begin
    Transmittan:= by * 1;
```

end;

Function rs_kon:boolean;

begin

reg.ah:=3;

reg.dx:=0;

intr(\$14,reg);

rs_kon:=((reg.ah and 1)=1);

end;

{.....Procedure kirim data ke port serial.....}

Procedure Kirimdata;

{Procedure datatx12;}

{Program Menampilkan_data_tertentu_di_file;(berdasarkan urutan record)}

{Uses Crt;}

Type

Minuman = Record

Nomor :byte;

Min1 :String[30];

Volu1 :byte;

Min2 :String[30];

Volu2 :byte;

Min3 :String[30];

Volu3 :byte;

Min4 :String[30];

Volu4 :byte;

Nama0 :String[30];

End;

Var

FileMinuman :File Of Minuman;

RecordMinuman :Minuman;

JumlahRecord :integer;

UrutanRecord :integer;

Lagi :Char;

I :integer;

data1, data2, data3, data4 :byte;

Begin

Assign(FileMinuman,'COCKTAIL.DAT');

Reset(FileMinuman);

JumlahRecord := FileSize(FileMinuman);

Lagi := 'Y';

While Uppcase(Lagi)='Y' do

Begin

clrscr;

gotoxy(5,6);

Write('Nomor record yang dicari (1 -',JumlahRecord:3,') ? ');

Readln(UrutanRecord);

writeln;

if (UrutanRecord <1) or (UrutanRecord > JumlahRecord) then

Writeln('Tidak ada nomor record ini !!!')

else

begin

```

        Seek(FileMinuman,UrutanRecord-1);
        Read(FileMinuman, RecordMinuman);
        gotoxy(5,8);
        Writeln('Nomor Record      : ',RecordMinuman.Nomor:5);
        gotoxy(5,10);
        Writeln('NAMA COCKTAIL           : ',RecordMinuman>Nama);
        gotoxy(5,13);
        Writeln('Bahan-bahannya :');

        GotoXY(5, 14);Write('Nama Minuman 1      : ',RecordMinuman.Min1);
        GotoXY(45,14);Writeln('Volume (ml) : ',RecordMinuman.Volu1:6);
        GotoXY(5, 15);Write('Nama Minuman 2      : ',RecordMinuman.Min2);
        GotoXY(45,15);Writeln('Volume (ml) : ',RecordMinuman.Volu2:6);
        GotoXY(5, 16);Write('Nama Minuman 3      : ',RecordMinuman.Min3);
        GotoXY(45,16);Writeln('Volume (ml) : ',RecordMinuman.Volu3:6);
        GotoXY(5, 17);Write('Nama Minuman 4      : ',RecordMinuman.Min4);
        GotoXY(45,17);Writeln('Volume (ml) : ',RecordMinuman.Volu4:6);

    end;

    gotoxy(5,20);
    Write('Ada lagi yang akan dicari (Y/T) ? '); Readln(Lagi);
end;

data1 := RecordMinuman.Volu1 ;
data2 := RecordMinuman.Volu2 ;
data3 := RecordMinuman.Volu3 ;
data4 := RecordMinuman.Volu4 ;

Close(FileMinuman);

begin
    tampak;
    window(36,3,73,32);
    textbackground(blue);
    clrscr;
    reg.ah:=0;
    reg.dx:=0;
    reg.al:=c3;    {baud rate = 4800 bit/s}
    intr($14,reg);
    index:=0;
    repeat
        if rs_kon Then
            begin
                reg.ah:=1; reg.dx:=0;    {mengirim data satu karakter}
                reg.al:=data1;
                intr($14,reg);
            end
        until rs_kon = 0;
    until rs_kon = 0;

asm
@salah:
    and ah,10000000b
    or  ah,ah
    jnz @salah
end;

    delay(1500);
    by:=reg.al;
    if by=13 then writeln    {=13 enter}

    else
        bufferS[index]:= reg.al;
        inc(index);
    end; writeln(index:2,' ',by); by:=0;
end;

```

```

        if rs_kon Then
        begin
            reg.ah:=1; reg.dx:=0;      {mengirim data satu karakter}
            reg.al:=data2;
            intr($14,reg);
asm
@salah:
    and ah,10000000b
    or  ah,ah
    jnz @salah
end;

        delay(1500);
        by:=reg.al;
        if by=13 then writeln    (=13 enter)
        else

            bufferS[index]:= reg.al;
            inc(index);
            writeln(index:2,'    ',by); by:=0;
        end;

        if rs_kon Then
        begin
            reg.ah:=1; reg.dx:=0;      {mengirim data satu karakter}
            reg.al:=data3;
            intr($14,reg);
asm
@salah:
    and ah,10000000b
    or  ah,ah
    jnz @salah
end;

        delay(1500);
        by:=reg.al;
        if by=13 then writeln    (=13 enter)
        else
            bufferS[index]:= reg.al;
            inc(index);
            writeln(index:2,'    ',by); by:=0;
        end;

        if rs_kon Then
        begin
            reg.ah:=1; reg.dx:=0;      {mengirim data satu karakter}
            reg.al:=data4;
            intr($14,reg);
asm
@salah:
    and ah,10000000b
    or  ah,ah
    jnz @salah
end;

        delay(1500);
        by:=reg.al;
        if by=13 then writeln    (=13 enter)
        else
            bufferS[index]:= reg.al;
            inc(index);
            writeln(index:2,'    ',by); by:=0;

```



```

end;

until keypressed;      {tambah readln : bila enter dilanjutkan}
readln;
window(1,1,80,25);

end;
end;

Procedure Tanya_Exit;
Var
  f:char;
begin
  GotoXY(30,21);Write(' tekan esc ');
  delay(2000);

  begin
    Textbackground(black);
    Textcolor(White);
    Tampak;
    Clrscr;
    exit;
  end;
end;

Procedure Lakukan_Pilihan(Pilih: integer);
Var
  X1,X2 : integer;
begin
  X1:= WhereX;
  X2:= WhereY;
  Move(Screen,ScreenSave,4000);
  Case Pilih of
    8: Kirimdata;
    9:  ;
    10: ;
    11: ;
    12: ;
    13: Tanya_exit;
  end;
  Move(ScreenSave,Screen,4000);
  GotoXY(X1,X2);
end;

Procedure KursorPilih;
Var
  Pilih : integer;
begin
  Pilih :=8;
  Repeat
    K:= Readkey;
    if ((Ord(K) = $0D) And (Pilih In[8,9,10,11,12,13])) Then
      begin
        Lakukan_Pilihan(Pilih);

      end
    else
      if K = #72 then    {panah atas}
      begin
        Baris := WhereY; Pilih := Baris-1;

```

```

        if Pilih <> 7 then
        begin
            Textbackground(black);
            Pilih_Tulisan(Pilih);
            Textbackground(Blue);
            Pilih_Tulisan(Baris);
            GotoXY(22,Pilih)
        end;
    end
else
    if K = #80 then
    begin
        Baris := WhereY; Pilih := Baris+1;
        If Pilih <> 14 then
        begin
            Textbackground(black);
            Pilih_Tulisan(Pilih);
            Textbackground(Blue);
            Pilih_Tulisan(Baris);
            GotoXY(22,Pilih);
        end;
    end;
    Until K = #27;
end;

{.....Main Program.....}
Begin

    GotoXY(22,8);
    Write('  Kirim Data ke Port Serial  ');

    GotoXY(22,13);
    Write('  Kembali ke menu Utama  ');

    CursorPilih;
    Textbackground(Black);
    Textcolor(White);
    Clrscr;

end.

```